



Universidad
Carlos III de Madrid

TRABAJO FIN DE GRADO

DESARROLLO DE UN ASISTENTE MULTIMODAL EDUCATIVO PARA DISPÓSITIVOS MÓVILES ANDROID

Autor: Adrián Muñoz Moreno

Tutor: David Griol Barres

Grado: Doble Grado en Ingeniería Informática y Administración
de Empresas

Madrid, 26 de septiembre de 2016

RESUMEN

El presente documento se corresponde con el Trabajo de Fin de Grado elaborado para concluir el Doble Grado en Ingeniería Informática y Administración de Empresas. El objeto de este trabajo se refiere al grado en Ingeniería Informática.

En él se ha explicado el proceso seguido para desarrollar una aplicación para dispositivos móviles enmarcada dentro del ámbito educativo siendo su objetivo ofrecer al usuario la capacidad de crear preguntas con respuestas y palabras clave para finalmente someterse a pruebas de conocimiento y contestar a preguntas de tipo test y de desarrollo, conociendo finalmente el resultado y viendo su progresión.

Para ello, ha sido necesario, en primer lugar, planificar el trabajo con fijación de fechas. El siguiente paso fue resolver el estado del arte, dónde se estudiaron plataformas *mobile learning*, concepto que engloba todas las aplicaciones y recursos que sirven para aprender a través de dispositivos móviles. Realizando un análisis exhaustivo sobre aplicaciones similares con el fin de evaluarlas y ver qué características son útiles para su implementación. Fue necesario conocer el estado de las herramientas disponibles en el mercado y cuáles sería conveniente utilizar, en este caso se escogieron: *Android Studio* para la codificación de la aplicación y *MySQL* para la creación de la base de datos.

Todo esto puede entenderse como la fase de planificación, como se explica a lo largo del documento.

Tras adquirir los conocimientos necesarios, se comenzó la fase centrada en el desarrollo de la aplicación. Utilizando manuales, tutoriales y videos a los que se hace referencia en cada punto, con el fin de poder completar esta fase.

Una vez concluido este, se realizaron pruebas en 2 plataformas: la misma herramienta dónde se implementó y un dispositivo móvil, encontrando satisfactorio el resultado de las mismas.

Antes de realizar las conclusiones y el trabajo futuro, se llevó a cabo una encuesta a la que contestaron 72 usuarios y cuyo fin era comprobar si el trabajo realizado se alineaba con sus expectativas y poder mejorarlo gracias a sus opiniones. En gran medida el trabajo realizado era bien recibido por el público y se anotaron sus deseos para versiones futuras de la aplicación.

Finalmente se realizaron las conclusiones, dónde se ha hecho un repaso de los aspectos más importantes del desarrollo, así como problemas encontrados durante este. Además, se enunció una línea futura de investigación influenciada mayoritariamente por, como ya se ha comentado, las opiniones directas de los encuestados.

Palabras clave: mobile learning, Android, sistema operativo, base de datos, test.

Índice de Contenidos

RESUMEN	2
CAPÍTULO 1. INTRODUCCIÓN	8
1.1 Motivación	8
1.1.1 Mobile learning	8
1.2 Objetivos	10
1.2.1 Objetivos del TFG	10
1.2.2 Objetivos de la aplicación.....	10
1.3 Planificación	11
1.3.1 Fase de planificación	11
1.3.2 Fase de desarrollo	11
1.3.3 Fase de documentación	12
1.4 Medios empleados	12
1.4.1 Software	12
1.4.2 Hardware	12
1.4.3 Lenguajes de programación	12
1.4.4 Manuales	12
1.5 Planificación temporal.....	13
1.6 Presupuesto	14
1.7 Estructura de la memoria.....	15
1. Introducción	15
2. Estado del arte	15
3. Descripción general de la aplicación desarrollada	15
4. Evaluación de la aplicación.....	15
5. Conclusiones y trabajo futuro	15
6. Glosario	15
7. Bibliografía	15
8. Anexo.....	15
CAPÍTULO 2. ESTADO DEL ARTE	16
2.1 Aplicaciones similares	17
2.1.1 App GoConqr	18
2.1.1.1 Ventajas GoConqr	19
2.1.1.2 Desventajas GoConqr	20
2.1.1.3 Conclusión del uso de GoConqr	20
2.1.2 App QuizMEOnline	20
2.1.2.1 Ventajas QuizMEOnline.....	22

2.1.2.2	Desventajas	22
2.1.2.3	Conclusión del uso de QuizMEOnline	22
2.1.3	App Quizlet	23
2.1.4	Ventajas Quizlet	24
2.1.5	Desventajas Quizlet.....	24
2.1.6	Conclusión del uso de Quizlet	25
2.1.7	Conclusión general sobre aplicaciones similares	25
2.2	Sistemas multimodales	25
2.2.1	Funcionamiento	26
2.3	Android.....	27
2.3.1	Android Studio.....	30
2.3.2	Proyecto en Android Studio	30
2.4	Bases de datos.....	32
2.4.1	SGBD – Sistema de Administración de Base de Datos	32
2.4.2	MySQL	33
CAPÍTULO 3.	DESCRIPCIÓN GENERAL DE LA APLICACIÓN DESARROLLADA	35
3.1	Presentación de la aplicación.....	36
3.1.1	Inicio de sesión/Registro	36
3.1.2	Insertar y visualizar preguntas	38
3.1.3	Realizar un TEST	40
3.1.4	Estadísticas y progresión del usuario	41
3.2	Arquitectura de la aplicación	42
3.2.1	Modelo	43
3.2.2	Vista.....	44
3.2.3	Controlador	44
3.3	La interfaz oral.....	45
3.3.1	Componentes de entrada.....	45
3.3.2	Componentes de salida	46
3.3.3	Ejemplo input-output aplicación a desarrollar.....	47
CAPÍTULO 4.	EVALUACIÓN PRELIMINAR DE LA APLICACIÓN	48
4.1	Entorno de pruebas.....	49
4.1.1	Emulador Android Studio	49
4.1.2	Dispositivo móvil	50
4.2	Pruebas realizadas.....	51
4.3	Encuesta a usuarios potenciales	57
4.3.1	Preguntas realizadas	57

4.3.2	Respuestas obtenidas y conclusiones extraídas	61
CAPÍTULO 5.	CONCLUSIONES Y TRABAJO FUTURO	66
5.1	Conclusiones.....	67
5.2	Trabajo Futuro.....	68
GLOSARIO	69
Bibliografía	72
ANEXO	74
1.	Contenidos en Inglés	74
1.1	ABSTRACT	74
1.2	INTRODUCTION	75
1.2.1	Motivation.....	75
1.2.2	Mobile learning	75
1.3	AIM	77
1.3.1	TFG aim.....	77
1.3.2	App aim	77
1.4	PLANNING	77
1.4.1	Planning phase	77
1.4.2	Development phase	78
1.4.3	Redaction phase	78
1.5	NEEDED RESOURCES	78
1.5.1	Software	78
1.5.2	Hardware.....	78
1.5.3	Programming languages.....	78
1.5.4	Manuals.....	78
1.6	Temporal planning	79
1.7	Budget	79
1.8	DOCUMENT STRUCTURE	80
1.9	GENERAL CONCLUSIONS	82
1.9.1	FUTURE LINES OF INVESTIGATION	83
2.	Prototipo JustInMind.....	84

Índice de Figuras

Figura 1. Diagrama de GANTT (planificación por semanas).....	13
Figura 2. App GoConqr: Vista de recursos destacado	18
Figura 3. App QuizMEOnline: Vista de quizzes.....	20
Figura 4. App Quizlet: Vista móvil principal	23
Figura 5. App Quizlet: Vista web principal	23
Figura 6. Evolución de versiones Android	27
Figura 7. Base de Datos distribuida	32
Figura 8. Vista de la aplicación: Pantalla de inicio.....	36
Figura 9. Vista de la aplicación: Pantalla de registro.....	36
Figura 10. Vista de la aplicación: Pantalla de selección	37
Figura 11. Vista de la aplicación: Inserción pregunta.....	38
Figura 12. Vista de la aplicación: Visualización preguntas	39
Figura 13. Vista de la aplicación: Selección tipo TEST	40
Figura 14. Vista de la aplicación: Estadísticas y progresión del usuario	41
Figura 15. Arquitectura MVC.....	42
Figura 16. Esquema relacional BBDD aplicación	43
Figura 17. Lista de pantallas en Android Studio.....	44
Figura 18. Componentes de entrada en un sistema de diálogo multimodal.....	45
Figura 19. Componentes de salida en un sistema de diálogo multimodal	46
Figura 20. Selección hardware emulador Android Studio	49
Figura 21. Selección software emulador Android Studio.....	49
Figura 22. Información del Software del dispositivo móvil.....	50
Figura 23. Versión de Android del dispositivo móvil.....	50
Figura 24. Preguntas encuesta: Introducción a usuarios	57
Figura 25. Preguntas encuesta: Comportamiento de los usuarios sobre aplicaciones móviles .	57
Figura 26. Preguntas encuesta: Hábitos del usuario durante el estudio	58
Figura 27. Preguntas encuesta: Contenidos aplicación educativa (1 de 2).....	59
Figura 28. Preguntas encuesta: Contenidos aplicación educativa (2 de 2).....	59
Figura 29. Preguntas encuesta: Datos estadísticos de los encuestados	60
Figura 30. Respuestas encuesta: SO más utilizado	61
Figura 31. Respuestas encuesta: Usuarios que no usan apps móviles	61
Figura 32. Respuestas encuesta: Recursos utilizados por los usuarios.....	62
Figura 33. Respuestas encuesta: Recomendaciones de mejora de los usuarios	64
Figura 34. Esquema pantallas del prototipo	84
Figura 35. Pantalla I Prototipo.....	85
Figura 36. Pantalla II Prototipo.....	85
Figura 37. Pantalla III Prototipo.....	86
Figura 38. Pantalla IV Prototipo	86
Figura 39. Pantalla V Prototipo	87
Figura 40. Pantalla VI Prototipo	88
Figura 41. Pantalla VII Prototipo	88
Figura 42. Pantalla VIII Prototipo	89
Figura 43. Pantalla IX Prototipo	89
Figura 44. Pantalla X Prototipo	90
Figura 45. Pantalla XI Prototipo	90

Índice de Tablas

Tabla 1. Ventajas e inconvenientes Mobile learning	9
Tabla 2.Presupuesto: Costes personal	14
Tabla 3.Presupuesto: Costes equipos y herramientas	14
Tabla 4.Presupuesto: Costes totales	14
Tabla 5. Ejemplo tabla de casos de prueba.....	51
Tabla 6. Caso de prueba 1: Registro en la aplicación	52
Tabla 7. Caso de prueba 2: Inicio de sesión en la aplicación.	52
Tabla 8. Caso de prueba 3: Inserción de pregunta.....	53
Tabla 9. Caso de prueba 4: Realización de TEST.	53
Tabla 10. Caso de prueba 5: Realización de pregunta abierta.....	54
Tabla 11. Caso de prueba 6: Edición de pregunta guardada.	54
Tabla 12.Caso de prueba 7: Visualización de progreso y estadísticas.	55
Tabla 13. Caso de prueba 8: La aplicación detecta la pantalla táctil.	55
Tabla 14. Caso de prueba 9: La aplicación detecta el teclado.	55
Tabla 15. Caso de prueba 10: La aplicación detecta el reconocimiento de voz.	56
Tabla 16. Caso de prueba 11: Interfaz intuitiva, amigable y usable.	56

CAPÍTULO 1. INTRODUCCIÓN

En este documento se recoge el desarrollo de un proyecto desde la idea inicial hasta su finalización como objeto de Trabajo de Fin de Grado (en adelante TFG). El trabajo se basa en la implementación de una aplicación para dispositivos cuyo sistema operativo sea Android y además de una interfaz táctil disponga de una oral.

La memoria presentada se divide en varios capítulos que se detallan en el último punto de este capítulo.

La temática seleccionada para la aplicación es educativa y el principal objetivo de ésta es ayudar a estudiantes a repasar los conocimientos que previamente han estudiado mediante el uso de tests que crearon ellos mismos mediante interfaces táctiles y orales.

1.1 Motivación

A la hora de seleccionar el tema del TFG, la motivación para escoger el desarrollo de una aplicación para un dispositivo móvil vino precedida de la demanda actual de contenidos tecnológicos. Todo lo que la gente quiere se encuentra a un clic y ese clic se realiza cada día con más frecuencia desde un dispositivo móvil.

Actualmente existen 4 sistemas operativos mayoritarios en el mercado (Android, iOS, Windows Phone, BlackBerry), 2 de ellos se reparten el 98% del mercado [1], estos son:

- Android toma alrededor del 75% del mercado.
- iOS se conforma con un 23%.

Es decir, 1 de cada 4 usuarios, posee Android en su dispositivo móvil como sistema operativo (en adelante SO). Debido a esto, se decidió que se podría llegar a un público más amplio si el desarrollo inicial de la aplicación se realizaba para dispositivos cuyo SO fuera Android.

Tras conocer qué se iba a estudiar, se debía pensar en cómo desarrollarlo y sobre qué aspecto trataría. Mientras que la creación de un juego sería muy complicada y no se tenía una idea sobre uno que pudiera mejorar los ya presentes en el mercado y, se quería realizar un proyecto que ayudara a los usuarios a realizar acciones cotidianas pero gracias a su dispositivo móvil, se encontró el concepto *mobile learning*.

1.1.1 Mobile learning

“Esta metodología educativa permite tanto al profesor como al alumno mantener un contacto constante en cualquier momento del día, fomentando con ello una educación individualizada y adaptándose a las necesidades del alumno en cada momento (todas las personas no se motivan en las mismas horas del día)..., y así con un largo etcétera de ventajas.”[2]

En la actualidad, las tecnologías han invadido la vida de los usuarios, desde los procesos más sencillos como hacer la compra pudiéndose realizar online, hasta los más complejos como tener una cita con el médico a través de *Skype*. Incluso, las tecnologías han cambiado la forma de enseñar y de estudiar; los recursos se suben a la red en lugar de tomar apuntes; los alumnos se enfrentan a test online en lugar de presenciales; los profesores se comunican a través de foros para resolver dudas a sus alumnos en lugar de realizar tutorías; entre un sinfín de novedades.

Todos estos procesos educativos en los que interfiere la tecnología se pueden llevar a cabo con un aprendizaje semipresencial, a lo que se llama *b-learning*, o con uno online, conocido como *e-learning*. Pudiendo el alumno decidir en qué momento estudiar y desde dónde quiere hacerlo. En muchas ocasiones, la gente tiene momentos de espera en los que no se dispone de nada más que un dispositivo móvil: desde la sala de espera del médico hasta el trayecto en tren al centro de estudios. Ambas son situaciones perfectas para aprender y se refieren a *mobile learning*.

Según A. Pisanty, L. Enriquez, L. Chaos – Cador, M. García, el término *mobile learning* aparece a finales de los 90 en EEUU gracias al uso de agendas electrónicas y en Europa no lo hace hasta 2001 con el proyecto [M learning](#) cuyo objetivo era mejorar las habilidades que fueran útiles en la vida cotidiana de personas que habían abandonado sus estudios, como matemáticas y comprensión lectora. Actualmente el proyecto sigue en pie y ayuda a grupos interesados en recibir educación no formal.

Desde finales de los 90 hasta el día de hoy, el número de usuarios que disponen de un dispositivo móvil ha aumentado exponencialmente y la edad de adquisición de uno se ha reducido notablemente. A pesar de que muchos usuarios solamente utilicen su teléfono para actividades de ocio, esa amenaza se puede convertir en oportunidad gracias a usar herramientas educativas. La enseñanza se debe adaptar a los recursos de los que dispone la sociedad y puede convertir el problema de los teléfonos en las aulas en la solución para abrir recursos que no aparecen en los libros de texto o incluso para acceder a libros de texto.

Entre todas las definiciones encontradas sobre *mobile learning*, la que más encaja con el objetivo del mismo es la ofrecida por e-ISEA en 2009:

“El mobile learning es una nueva forma de educación creada a partir de la conjunción entre el e-learning y la utilización de los smart devices/dispositivos móviles inteligentes (pda’s, smartphones, iPods, pocket PCs, teléfonos móviles 3G, consolas, etc.), y que se fundamenta en la posibilidad que nos ofrecen estos nuevos dispositivos, de combinar la movilidad geográfica con la virtual, lo cual permite el aprender dentro de un contexto, en el momento en que se necesita y explorando y solicitando la información precisa que se necesita saber.”

Para conocer las principales ventajas e inconvenientes del su uso, se han resumido los puntos a favor y los puntos en contra en la siguiente tabla:

Ventajas	Inconvenientes
Aprendizaje en cualquier momento y lugar.	Pantallas pequeñas de los dispositivos.
Facilidad comunicación alumno-profesor.	Rechazo de nuevas tecnologías.
Accesibilidad a gran número de recursos.	Distracción de los usuarios.
Aprendizaje colaborativo (compartición de contenidos).	Almacenamiento en el dispositivo.
Incentivación al estudio.	Alto consumo batería del dispositivo.
Integración de la enseñanza y nuevas tecnologías.	Actualizaciones constantes.

Tabla 1. Ventajas e inconvenientes Mobile learning

Las ventajas que se pueden conseguir con el uso de esta metodología son infinitas, mientras que los inconvenientes tratan sobre todo de problemas con el dispositivo que se podrían solventar con tarjetas extraíbles para el problema del almacenamiento o baterías externas para el del consumo de la batería.

El *mobile learning* es el proceso educativo del futuro ya que ofrece a los usuarios un amplio abanico de ventajas. Actualmente no se concibe un mundo sin tecnología y la enseñanza debe adaptarse a ello. En las universidades ya se ha implantado un sistema para que facilite la comunicación entre alumnos y profesores. En los colegios todavía se está migrando a ese cambio. Ahora solo falta que los propios estudiantes utilicen este sistema independientemente sin necesidad de una entidad que lo ponga a su disposición. La forma de usarlo podría ser la descarga de aplicaciones que faciliten el estudio.

Por ello, la aplicación que se ha decidido desarrollar ofrece a los estudiantes el poder repasar sus conocimientos en cualquier lugar y a cualquier hora, sin necesidad de cargar con libros y apuntes, simplemente seleccionando la aplicación, cargando preguntas y sometiéndose a pruebas.

1.2 Objetivos

Esta sección se divide en 2: objetivos del TFG y objetivos de la aplicación desarrollada.

1.2.1 Objetivos del TFG

Al seleccionar dicho TFG y tener la primera reunión con el tutor que lo ha supervisado, los objetivos del trabajo a desarrollar eran claros:

- Estudio de aplicaciones similares en el mercado de los dispositivos móviles: este estudio se realizaría con el fin de encontrar qué características tienen las aplicaciones del tipo pregunta-respuesta para ver qué diferenciación se puede ofrecer al usuario para que elija esta aplicación en lugar de una ya existente en el mercado.
- Aprender a usar nuevas herramientas relacionadas con el desarrollo de aplicaciones móviles: con el fin de desarrollar la aplicación, aprender a usar herramientas necesarias para ello:
 - *Android Studio*: plataforma en la que se escribe el código de la aplicación (interfaz de usuario, lógica de negocio y persistencia)
 - *Justinmind*: prototipado de interfaces que pueden ser visualizadas en dispositivos móviles.
 - *MySQL*: desarrollo de base de datos para guardar información que requiera un espacio amplio.
- Orientar lenguajes conocidos a la programación Android: orientar los lenguajes estudiados durante el grado realizado a la programación en Android. Etiquetas *XML*, lenguaje *Java* y consultas *SQL*.
- Realizar una aplicación que pueda ser usada por usuarios móviles: desarrollar una aplicación que pueda ser descargada y utilizada por usuarios de dispositivos móviles con el fin de repasar sus conocimientos.

1.2.2 Objetivos de la aplicación

Los objetivos que se quieren lograr al lanzar la aplicación en cuestión al mercado son los siguientes:

- Ofrecer al usuario una alternativa simple y útil a las aplicaciones existentes: ofrecer a los usuarios una aplicación sencilla de utilizar gracias a una interfaz intuitiva. Que ayude a poner a prueba sus conocimientos sobre temas que quieren repasar de una forma fácil pudiendo hacerlo desde cualquier lugar y a cualquier hora.

- Colaborar en la integración del aprendizaje mediante el uso de nuevas tecnologías: facilitar al usuario una aplicación de estudio sencilla para que la integración en la sociedad de las aplicaciones educativas se convierta en una realidad y progrese con mayor rapidez.
- Accesibilidad táctil y oral: todas las pantallas y funcionalidades están disponibles para una interfaz táctil y además la inserción de preguntas y respuestas pueda realizarse a través de la voz con una interfaz oral.

1.3 Planificación

La realización de este proyecto se divide en 3 fases: planificación, desarrollo y documentación.

1.3.1 Fase de planificación

Fase de establecimiento de objetivos, tanto del TFG como de la aplicación, así como los tiempos que se deben cumplir en la realización del proyecto. También engloba un estudio previo al desarrollo de las aplicaciones similares en el mercado y prototipado de interfaces. Se explica de forma resumida en el siguiente listado:

- Establecimiento de objetivos para conocer el alcance de la aplicación y del TFG.
- Establecimiento de tiempos mediante un diagrama de GANTT que se puede encontrar en la Figura 2 de la sección [1.4 Planificación Temporal](#).
- Selección de herramienta a utilizar para el desarrollo de la aplicación: *Android Studio* frente a *Eclipse* con *plugin* de Android. Esta decisión se tomó como parte del objetivo de utilizar herramientas no vistas a lo largo del grado.
- Estudio de aplicaciones similares en el mercado: con el fin de “copiar” las funcionalidades buenas de cada aplicación e incluir alguna funcionalidad que diferencia a la aplicación a desarrollar de las que ya están disponibles en el *Market* de Android.
- Prototipado de interfaces: diseño de las distintas pantallas que podrá ver el usuario en la aplicación. Siguiendo el objetivo de utilizar herramientas nuevas, se escogió la herramienta *Justinmind* para realizar el prototipo de las pantallas, dicho prototipo se incluye en la sección del Anexo [8.2 Prototipo JustInMind](#).

1.3.2 Fase de desarrollo

Fase de desarrollo la aplicación que se va a ofrecer a los usuarios para que puedan poner a prueba sus conocimientos. Además del desarrollo, engloba una serie de pruebas realizadas para comprobar la robustez de la aplicación. El desarrollo de la aplicación se puede dividir en 3 tareas nucleares:

- Desarrollo de las interfaces en *Android Studio* mediante el uso de etiquetas *XML*.
- Desarrollo de la lógica de negocio en *Android Studio* utilizando conocimientos adquiridos durante el grado en lenguaje *Java*.
- Desarrollo de persistencia en *Android Studio* utilizando conocimientos adquiridos durante el grado en lenguaje *SQL*. Además de implementar las distintas consultas e inserciones/borrados en la herramienta cita, se debe montar una base de datos en la aplicación *MySQL* ya que sin ella no se podrán realizar operaciones contra la base de datos.

Las pruebas realizadas para testar la aplicación se detallarán en sección [4.2 Pruebas realizadas](#).

1.3.3 Fase de documentación

Fase en la que se documenta el estudio previo realizado, estado de la aplicación desarrollada y problemas encontrados durante el desarrollo, ofreciendo finalmente una conclusión sobre el proyecto.

Esta fase se realizará al acabar el desarrollo de la aplicación, aunque se seguirán las notas tomadas y la experiencia adquirida durante el desarrollo de la aplicación que es el núcleo central del TFG.

1.4 Medios empleados

En este apartado se detallan las herramientas software y hardware necesarias para la realización completa del TFG, los lenguajes de programación con los que se ha trabajado a la hora de trabajar en el desarrollo de la aplicación y manuales consultados para poder completar el proyecto en su totalidad. A continuación se detallan diferenciando en software, hardware, lenguajes de programación y manuales.

1.4.1 Software

Las herramientas software utilizadas son:

- Sistema operativo Windows 10 → plataforma sobre la que descargar las herramientas.
- Microsoft Office Word 2016 → escritura de la memoria.
- Justinmind → prototipado de las interfaces.
- GANTT Project → fijación de establecimiento de tiempos.
- Android Studio → desarrollo de interfaces, lógica de negocio y persistencia.
- MySQL → creación de base de datos.
- Notepad ++ → visualización del código.

1.4.2 Hardware

Los dispositivos hardware utilizados son:

- Portátil Lenovo ideapad y410p → dispositivo sobre el que montar el escritorio de trabajo
- Teléfono móvil LG G3S → pruebas de la aplicación.

1.4.3 Lenguajes de programación

Los lenguajes de programación empleados son:

- XML → desarrollo de interfaces.
- Java → desarrollo de lógica de negocio.
- SQL → desarrollo de persistencia y base de datos.

1.4.4 Manuales

Los manuales consultados durante el desarrollo son:

- TFG Adrián Muñoz Moreno ADE → estructura de un TFG y cómo hacer un presupuesto.
- Manual: "Curso Android Studio" [19] → cómo programar en Android Studio.
- Diversas páginas web contenidas en las referencias bibliográficas.

1.5 Planificación temporal

La idea inicial del proyecto surge en septiembre de 2015 al seleccionar como objeto de TFG “DESARROLLO DE UN ASISTENTE MULTIMODAL EDUCATIVO PARA DISPÓSITIVOS MÓVILES ANDROID” donde tras la primera reunión que tuvo lugar el 29 de aquel mes se especifica que la aplicación sobre la que se trabajará será una de preguntas-respuestas en la que el usuario podrá incluir sus propias preguntas y autoevaluarse.

La reunión inicial fue a finales de septiembre ya que el TFG debía ser seleccionado y matriculado en el primer cuatrimestre, a pesar de esto y debido a tener durante los 2 cuatrimestres del año asignaturas y el TFG del otro grado del doble grado por realizar, que se presentó a principios de Julio, se pospuso el proyecto de informática hasta verano y así poder dedicar enteramente el tiempo una vez finalizadas el resto de obligaciones universitarias. Por ello, el tiempo real que se expone en el siguiente diagrama de GANTT engloba desde el 1 de julio hasta el 26 de septiembre de 2016.

Lo primero que se realiza es un estudio previo del campo en el que se iba a lanzar la aplicación. Tras esto, se diseñaron las interfaces y se estructura como va a ser la aplicación antes de desarrollar el código. Una vez se finaliza con ello, se comienza con el desarrollo. Al término de éste, se realizan distintas pruebas para finalmente escribir la memoria.

A continuación, se muestra la planificación del proyecto:

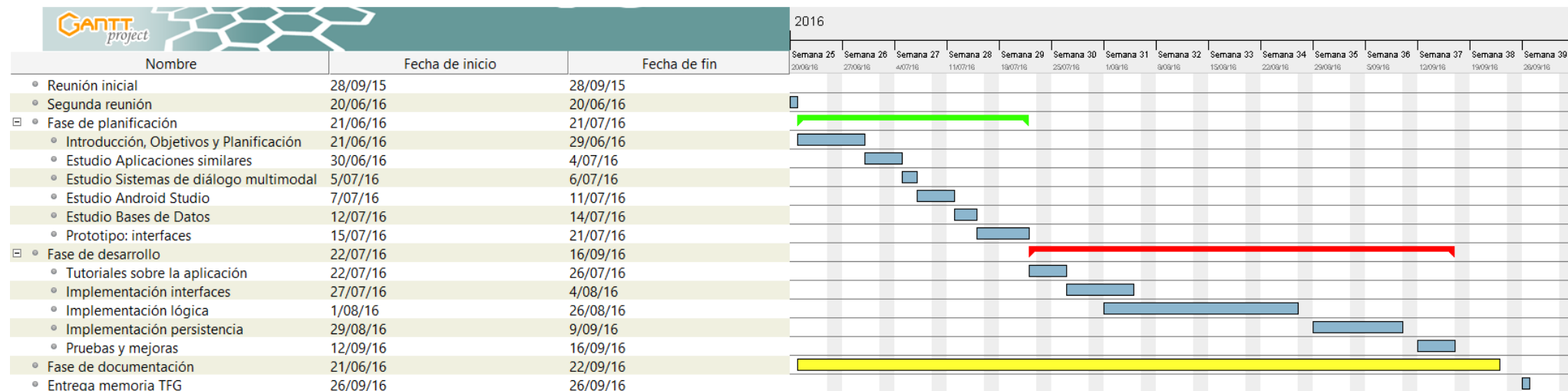


Figura 1. Diagrama de GANTT (planificación por semanas)

1.6 Presupuesto

En la tabla que se muestra a continuación figura el coste por hora de cada miembro que va a colaborar en el proyecto, así como el número de horas que va a trabajar en él. En la última columna, se encuentra el coste de cada trabajador en el proyecto. Para hallar el coste por hora se ha realizado una búsqueda por internet para ver el coste en el mercado de conseguir uno, mientras que para estimar el número de horas se ha revisado el GANTT y el tiempo invertido en el desarrollo del TFG.

Categoría	Coste por hora (€)	Número de horas	Coste
Jefe de proyecto	40,00 €	40	1.600,00€
Analista	25,00 €	150	3.750,00€
Programador	20,00 €	250	5.000,00€
Gestor de calidad y pruebas	15,00 €	40	600,00€
	Total	480	10.950,00€

Tabla 2.Presupuesto: Costes personal

Así como la tabla anterior mostraba el coste del personal que realizaría el proyecto, a continuación se muestra una tabla con el coste de los equipos y herramientas utilizadas para el desarrollo:

Equipo/herramienta	Coste	Vida útil (meses)	Tiempo de uso (meses)	Coste imputable
PC Portátil lenovo ideapad y410p	908,99€	36	3	75,749€
Teléfono móvil LG G3S [17]	121,18€	24	3	15,148€
Paquete Windows Office [18]	149,00 €	12	3	37,25€
Android Studio	0,00€	--	3	0,00€
MySQL Workbench	0,00€	--	3	0,00€
JustInMind (versión de prueba)	0,00€	--	3	0,00€
Inscripción Play Store	22,40€	--	--	22,40
			Total	150,547€

Tabla 3.Presupuesto: Costes equipos y herramientas

Para los equipos y herramientas se debe imputar al proyecto únicamente los costes amortizados durante su uso ya que su uso no es exclusivo del proyecto una vez este haya finalizado. Se ha seleccionado una amortización lineal para su imputación, es decir, los equipos y herramientas se amortizarán la misma cantidad cada año de su vida útil:

$$\text{Coste imputable} = \frac{\text{Coste}}{\text{Vida útil}} \times \text{tiempo de uso}$$

Con costes de personal y de equipo, se obtiene la siguiente tabla que muestra los costes totales:

Procedencia del coste	Coste
Personal	10.950,00€
Equipos/herramientas	150,547€
Total	11.078,147€

Tabla 4.Presupuesto: Costes totales

La forma de conseguir beneficios del proyecto es la venta de la aplicación en el Play Store por valor de 1,00€, de los cuales que Google tiene un 30%, por cada usuario que compre la aplicación, se recibirán 0,70€. El punto muerto en el que se habrá cubierto el desembolso inicial de 11.078,147€ será al superar las 15.825 descargas.

1.7 Estructura de la memoria

En las siguientes líneas, se incluye un resumen de cada capítulo de la memoria con el fin de mostrar la estructura de la misma.

1. Introducción

En este punto se especifica cómo es el TFG: que motivó al estudiante, los objetivos que se pretenden conseguir, la planificación que se ha seguido a la hora de realizar este proyecto, los medios empleados y el presupuesto necesario para completarlo.

2. Estado del arte

Se explican las aplicaciones similares que existen en el mercado, las ventajas y desventajas de cada una. Se expone un estudio sobre los sistemas multimodales actuales y, finalmente, se estudian las herramientas que se van a usar en el desarrollo así como las distintas bases de datos existentes seleccionando una para implementar en ella la persistencia de la aplicación.

3. Descripción general de la aplicación desarrollada

Se muestran las distintas pantallas de la aplicación y se explican las funcionalidades disponibles en cada una. Se especifica la arquitectura utilizada para el desarrollo y el esquema relacional de la base de datos. Finalmente se profundiza en los componentes de una interfaz oral y cómo funciona.

4. Evaluación de la aplicación

Se especifican una serie de pruebas que debería pasar la aplicación para tener la robustez deseada explicando cómo se han realizado y si el resultado ha sido positivo. Además, se ha incluido una encuesta posterior al desarrollo para comprobar si la aplicación cumple las expectativas del público y trazar una línea de trabajo futuro.

5. Conclusiones y trabajo futuro

Se analizan los resultados obtenidos y se expone una conclusión a modo de resumen sobre estos resultados. Se genera una línea de investigación en la que seguir a raíz del trabajo realizado.

6. Glosario

Explicación de términos de acrónimos y tecnicismos que se han mencionado a lo largo de la memoria.

7. Bibliografía

Se citan todos los medios, ya sean digitales o físicos, de los que se ha obtenido información para la realización del proyecto.

8. Anexo

En este apartado se incluye toda la información importante referente al proyecto y que no pertenezca a ninguno de los apartados anteriores: como el resumen de contenidos en inglés o el prototipo diseñado.

CAPÍTULO 2. ESTADO DEL ARTE

La siguiente frase encontrada en un manual para realizar un plan de empresa define muy bien la importancia que tiene la definición de este capítulo en la memoria:

“La clave del éxito necesariamente proviene de un buen análisis de partida, es un requisito para elaborar un buen plan de empresa y alcanzar el triunfo.” (Miranda Oliván, 2004)

Y es que, como se lee en ella, la clave del éxito proviene de un buen análisis de partida, ya no solamente para un plan de empresa, sino para cualquier proyecto ante el que se quiera obtener un buen resultado o, como termina la cita, alcanzar el triunfo.

Haciendo un análisis exhaustivo del ámbito en el que se quiere incluir la aplicación, resulta más sencillo ver cuáles son los puntos fuertes con los que cuenta el proyecto y sobre cuáles se debe incidir para mejorar.

El proyecto que se expone es una aplicación móvil para dispositivos cuyo sistema operativo es Android y que, además de entrada táctil mediante la pantalla del dispositivo, extenderá su capacidad a reconocimiento de voz. La principal funcionalidad de la aplicación será ofrecer al usuario la capacidad de autoevaluarse mediante preguntas que previamente él mismo cargó en el sistema. Por ello, en las siguientes líneas se analizarán:

- Aplicaciones similares que existen en el mercado: aplicaciones que permitan incluir preguntas-respuestas al usuario y autoevaluarse.
- Sistemas de diálogo multimodal: qué son y cómo funcionan.
- Historia de Android: nacimiento y evolución en versiones del sistema operativo más utilizado.
- Android Studio: herramienta en el que se va a desarrollar el código necesario para que la aplicación funcione correctamente.
- Bases de datos existentes: base de datos que se utiliza en el desarrollo de la aplicación.

2.1 Aplicaciones similares

La aplicación, como ya se ha comentado a lo largo del TFG tiene como objetivo ayudar a los usuarios a autoevaluarse de sus conocimientos con preguntas que previamente han cargado ellos mismos en la base de datos de la aplicación. Actualmente, este servicio lo ofrecen diversas apps que podemos encontrar de forma online o en las tiendas de aplicaciones de los dispositivos móviles. Haciendo una búsqueda por la red se han identificado las siguientes aplicaciones cuyo servicio es similar al que se quiere ofrecer y que por tanto son competidores directos. La investigación se fija en ellas estudiándolas y probándolas para ver cuáles son sus puntos fuertes y en qué aspectos flojean, con el fin de imitar sus funcionalidades estrella y mejorar o incluir funcionalidades que demanden los usuarios y no estén implementadas.

A continuación se enuncian y resumen brevemente las principales aplicaciones encontradas [3]:

- Google Forms: permite crear formularios y compartirlos con otros usuarios. No se centra en la creación de formularios, sino que es una funcionalidad más de todas las que ofrece, por lo que no está especializado en ello ni enfocado al ámbito educativo. Se crean formularios pero no tests al uso como los que se pretenden ofrecer. Se accede desde el siguiente [enlace](#).
- Quizbean: permite generar cuestionarios de la longitud deseada, con fotos y explicaciones. **Facilita la autoevaluación de los alumnos y les proporciona un informe con sus resultados, esto es algo interesante y que puede resultar útil observar para implantarlo en la aplicación.** No dispone de aplicación móvil, simplemente es un recurso web y solo tiene versión en inglés, pudiendo ser una barrera por 2 motivos; 1. Al no tener aplicación móvil, su acceso no es tan rápido como el de tener descargada la aplicación en el dispositivo; 2. Estudiantes que no dispongan de un nivel de inglés suficiente podrían no saber usarla o simplemente no usarla por tener el hándicap del idioma. Se encuentra [aquí](#).
- Blubbr: página web dónde se enlazan vídeos de plataformas como YouTube y permite crear test al finalizar este con el fin de que los usuarios contesten a preguntas relacionadas con el mismo. Puede usarse para el ámbito educativo, pero es algo para estudiantes más pequeños o con estudios relacionados con el cine o la música. Accede para navegar desde este [enlace](#). (Versión en inglés)
- Gnowledge: permite crear, publicar, compartir, y realizar tests, actividades y deberes desde su versión web. Su funcionalidad es muy completa abarcando más terreno que la elaboración de tests y es una forma de comunicación y ayuda al estudio de profesores y alumnos. [Enlace directo](#). (Solo en inglés)
- Quiz me online: esta herramienta solo dispone de versión web y tiene una funcionalidad parecida a la explicada en el punto anterior (Gnowledge). Esta aplicación será estudiada en profundidad en este punto ya que es de las más completas. Se puede acceder desde [aquí](#).
- Quizlet: disponible tanto de versión online como de aplicación para iOS y Android. Permite crear cuestionarios interactivos y da acceso a un enorme repositorio de tests online. A través de ella, tanto alumnos como profesores pueden crear formularios y juegos educativos y retar a otros a resolverlos. Ésta, al igual que la anterior, será estudiada en profundidad al tener muchas similitudes con el proyecto al que se desea llegar. Acceso desde este [enlace](#).

- **GoConqr** (Antiguo ExamTime): esta aplicación sería el principal competidor ya que tiene un comportamiento muy similar al deseado, aunque la elaboración de contenido solo se puede hacer a través de su versión web y no desde la propia aplicación móvil. Es en la que más usuarios registrados existen entre las estudiadas y por tanto, en la que más contenidos se comparten. Se estudiará en profundidad en el siguiente punto. Acceso directo [aquí](#).

Esta app permite:

- Visualizar y compartir Diapositivas
- Mapas Mentales
- Fichas de Memoria
- Apuntes Online
- Tests creados con la versión web
- Compartir recursos con compañeros

Tras este amplio estudio en el que se han buscado aplicaciones con un comportamiento similar al deseado, se han seleccionado las 3 que más similitudes en funcionalidad tienen con la aplicación que se va a desarrollar: GoConqr, QuizMEOnline y Quizlet. Como ya se dijo, se estudiarán sus fortalezas y debilidades con el fin de imitar lo bueno de cada una e incluir las funcionalidades de las que no dispongan.

2.1.1 App GoConqr

Para realizar un estudio sobre las funcionalidades de la aplicación, su presentación y su funcionamiento, se ha hecho uso de ella como si se fuera un usuario novato.

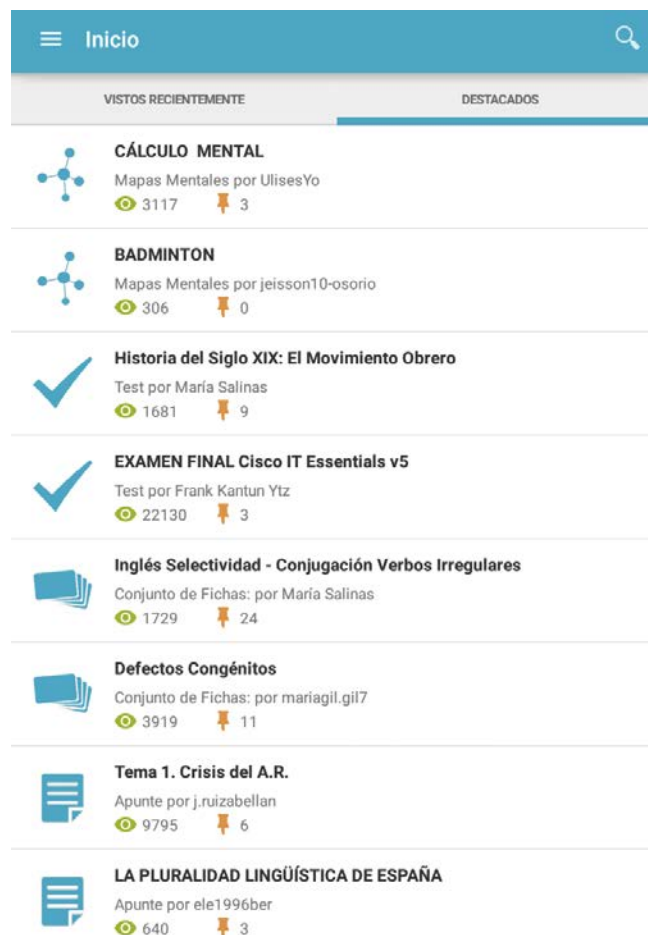


Figura 2. App GoConqr: Vista de recursos destacado

Una vez se ha descargado la aplicación, se abre para estudiar lo que se ha mencionado anteriormente.

Al abrir la aplicación se observa que se permite el registro desde Facebook, Google + o crear una cuenta nueva desde el e-mail. Una vez se ha completado el registro, el usuario se encuentra con contenidos compartidos por otros usuarios a los que se puede acceder sin problemas clasificados entre “Destacados” y “Vistos Recientemente”. Siendo posible buscar entre ellos y establecer un filtro de búsqueda.

El menú de usuario se divide en:

- Inicio: lleva al usuario a la pantalla principal que se observa tras iniciar sesión.
- Asignaturas: donde se pueden ir creando carpetas en las que guardar recursos propios o compartidos por otros usuarios. Dentro de cada asignatura se pueden crear subcarpetas.
- Destacados: aparecen los recursos compartidos por otros usuarios que más éxito están teniendo entre el resto de usuarios de la aplicación
- Visto Recientemente: los últimos recursos que ha abierto el usuario de más reciente a más antiguo.
- Ajustes: esta pantalla ofrece funcionalidades tales como: cerrar sesión, cambiarla dirección de correo electrónico, ver la versión de la app, etc.

Los recursos que se pueden crear o visualizar se dividen en categorías que se diferencian por iconos. Las categorías son; mapa conceptual; test; conjunto de fichas; y apuntes.

Tras este recorrido desde la vista de un usuario novato, se identifican las principales características de la aplicación, tanto favorables como desfavorables. Se exponen a continuación.

2.1.1.1 Ventajas GoConqr

- Es una aplicación intuitiva, permite al usuario navegar sin necesidad del uso de un tutorial ni haciendo que se sienta perdido y abandone la app por no saber cómo usarla.
- Su interfaz es amigable, mediante el uso de tres colores (azul, blanco y negro) se crean todos los menús necesarios. No provocando en el usuario un rechazo ante ella.
- Dispone de una gran cantidad de recursos y en gran variedad de ámbitos. Es una aplicación conocida y que dispone de una amplia cartera de usuarios activos.
- Permite puntuar los recursos para que los mejores sean vistos por más usuarios al ser clasificados en la vista de todo el mundo al iniciar la aplicación.
- Se pueden compartir los recursos con aplicaciones de mensajería o redes sociales externas a la aplicación mediante un link al documento.
- Es posible guardar los recursos ajenos (concepto pinear) por asignaturas y temas como si se tratasen de contenidos propios.
- Cada recurso dispone de una función que muestra contenidos relacionados.
- Se asignan etiquetas al subir cada recurso para una clasificación más concreta y poder acceder desde búsquedas tanto genéricas como exhaustivas.

2.1.1.2 Desventajas GoConqr

- No existen comentarios, mensajes ni se puede acceder a otros usuarios para preguntar dudas sobre sus contenidos o establecer vínculos para poder acceder a todos los recursos creados por un único usuario. Falla el componente social.
- Los recursos solo se pueden crear a través de la versión web.
- Los contenidos de los tests no se pueden mezclar con los de otros tests para evaluarte sobre varios temas de una misma asignatura o hacer un repaso general de varias asignaturas a la vez.
- Las asignaturas no se muestran si no tienen ningún contenido en su interior.
- No dispone de estadísticas ni guarda resultados sobre tests anteriores.

2.1.1.3 Conclusión del uso de GoConqr

Como conclusión, es una aplicación fácil e intuitiva que dispone de muchos usuarios siendo esta la principal fortaleza ya que una aplicación social sin usuarios pierde notoriedad y contenidos. Acierta en la organización gracias a etiquetas, relaciones y carpetas. Para competir contra ella habría que imitar sus ventajas, haciendo una aplicación amigable para el usuario, intuitiva y limitar los colores a una paleta de 3 similares para no producir una sensación de rechazo. Pero sobre todo, el desarrollo se debe centrar en aspectos de los que no dispone GoConqr, como poder crear contenidos desde la app móvil y que los resultados obtenidos por cada usuario se guarden y se creen estadísticas, siendo estas dos funcionalidades añadidas muy percibidas por los usuarios a nivel de comodidad y de evaluación de resultados.

2.1.2 App QuizMEOnline

Esta aplicación solo dispone versión web y en inglés. Para comprobar las funcionalidades que ofrece se ha hecho un recorrido a través de su sitio web.

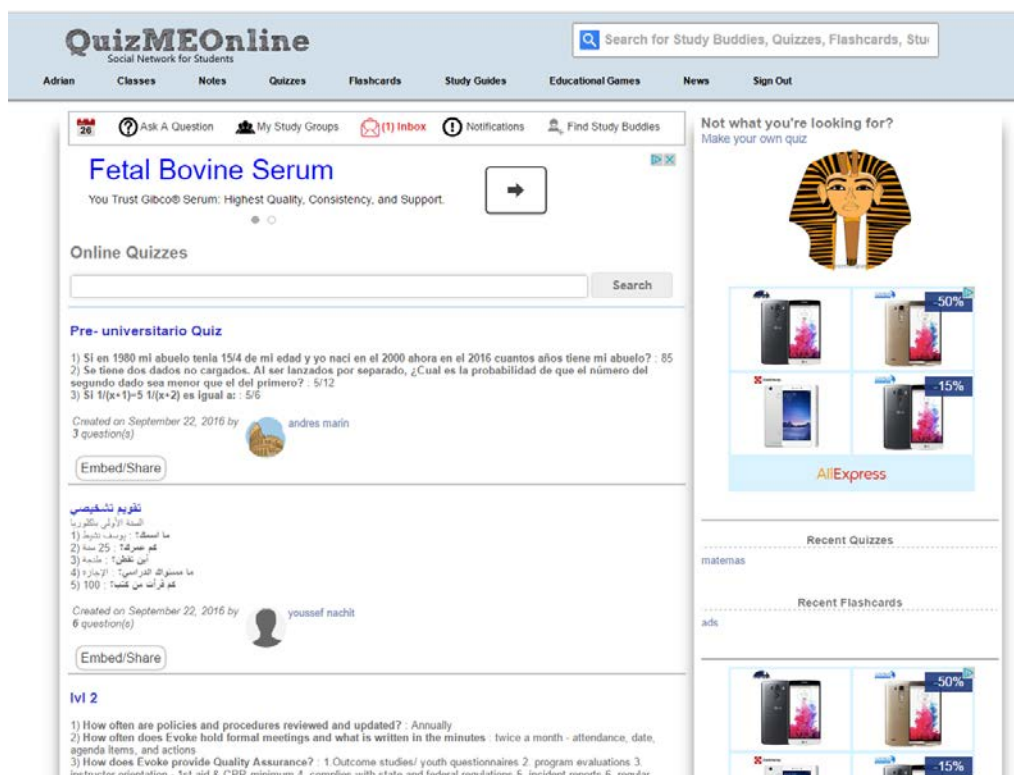


Figura 3. App QuizMEOnline: Vista de quizzes

Se puede visualizar y usar todo el contenido sin necesidad de registrarse, excepto para creación de contenido que sí que hay que registrarse. Este registro puede hacerse desde el e-mail o desde Facebook.

Dispone de una barra de navegación dividida en:

- Home: página principal que muestra novedades, temas en los que se pueden encontrar tests, principales funcionalidades que ofrece el sitio como crear un test, una guía de estudio, etc. Un tablón de nuevos miembros. Permite registrarse en una clase mediante un código que tendrá que proporcionar el administrador de esa clase.
- Quizzes: permite crear un nuevo quizz, ver quizzes propios, buscar quizzes y ver una lista de quizzes por tema. Al crear un quizz, se permite seleccionar si la pregunta es de Verdadero/Falso o de selección múltiple.
- Flashcards: permite crear tarjetas de estudio, ver tarjetas propias, buscar tarjetas y ver una lista de tarjetas por tema. Las tarjetas tienen 2 caras para poder poner la pregunta en un lado y la respuesta en el otro con el fin de estudiar mediante memoria y no por respuestas que ofrezca el programa.
- Study Guides: permite crear una guía de estudio, consultar las propias, ver el grupo de estudio, la clase, buscar guías y ver una lista de guías por tema. Al crear una guía de estudio, se añade el contenido a estudiar y las anotaciones que quiera el autor para ayudar al estudio del usuario.
- Educational Games: esta funcionalidad ofrece al usuario 3 juegos: ahorcado; ser más rápido que las flashcards que aparecen; y un juego de memoria.
- Teacher Solutions: disponible en el rol de profesor.
- News: noticias interesantes para estudiantes.
- Contact Us: formulario para poder enviar mensajes al equipo de la página web.

El menú de usuario se divide en:

- Ajustes de cuenta: permite al usuario cambiar sus datos personales
- Ajustes de privacidad: permite al usuario configurar parámetros como quien puede enviarle mensajes o quien puede ver y guardar los contenidos creados por el mismo.
- Buscador de compañeros: permite buscar antiguos compañeros de clase mediante nombre y escuela.
- Panel de compañeros: lista con compañeros de clase.
- Mensajes: pestaña que permite enviar y recibir mensajes de otros usuarios.
- Grupos: lista de grupos a los que pertenece el usuario.
- Invitar amigos: se puede invitar a algún amigo a la página introduciendo su e-mail.

Los contenidos se dividen principalmente en *quizzes*, *flashcards* y *study guides*.

Al no ser una aplicación móvil y ser una página web, ofrece más funcionalidades. Tras haber accedido a ella como un usuario se comentan a continuación sus características.

2.1.2.1 Ventajas QuizMEOnline

- Dispone de una gran cantidad de contenidos que ofrece al usuario una amplia gama de posibilidades.
- Buena estructuración y nombres intuitivos que permiten al usuario conocer en todo momento que es lo que va a realizar.
- Permite reencontrar a antiguos o actuales compañeros de clase para poder compartir material para favorecer el estudio.
- Permite chatear con otros usuarios para conocer como han estudiado ellos los contenidos. Además esta funcionalidad permite consultar/resolver dudas.
- Es posible guardar los contenidos de otros usuarios siempre y cuando el otro usuario acepte la petición.
- Dispone de juegos para que la autoevaluación sea más amena.

2.1.2.2 Desventajas

- No dispone de aplicación móvil.
- No dispone de estadísticas ni guarda resultados sobre tests anteriores.
- La interfaz produce cansancio al navegar durante varios minutos por ella debido a los colores y los contenidos flash en movimiento.
- Abuso de publicidad. En cada pantalla existen más de 2 anuncios.

2.1.2.3 Conclusión del uso de QuizMEOnline

Es una aplicación web que ofrece a los estudiantes diversas posibilidades pero no dispone de una versión móvil, un punto que es muy demandado actualmente para acceder a cualquier contenido desde el dispositivo portátil sin tener que buscar en el navegador. Como punto fuerte, se observa que hay varios tipos de recursos sobre los que crear contenidos, además de poder acceder a los ya creados por otros usuarios. Gana en el aspecto social. En el lado contrario, como punto débil, tiene una interfaz que no es amigable con el usuario ya que produce cansancio al navegar por su web debido a la cantidad de colores que utiliza y la excesiva publicidad que se encuentra en cada página.

2.1.3 App Quizlet

Se ha seguido el mismo método que en las anteriormente mencionadas: usar la app como si se trata de un usuario novato.

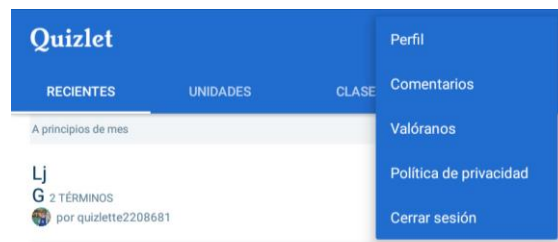


Figura 4. App Quizlet: Vista móvil principal

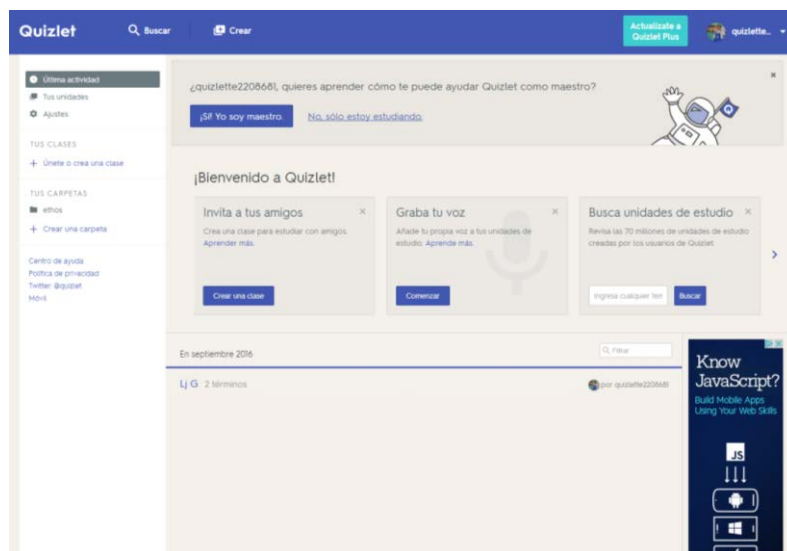


Figura 5. App Quizlet: Vista web principal

Quizlet, igual que App GoConqr permite el registro desde Facebook, Google + o crear una cuenta nueva desde el e-mail. La aplicación divide en 4 la barra de navegación:

- Recientes: ver los contenidos vistos en un corto espacio de tiempo anterior.
- Unidades: muestra las unidades de estudio del usuario. Desde aquí se podrán crear diversas unidades añadiéndole un título y seleccionando los idiomas en los que está disponible y si se comparte con todos los usuarios o es exclusivo del usuario que lo crea. Dentro de una unidad el usuario encuentra las siguientes opciones:
 - Fichas: fichas creadas previamente con contenidos a estudiar. Una ficha se compone de término + definición.
 - Aprender: expone la definición de una ficha y pregunta el término.
 - Combinar: juego de memoria que busca que el usuario una cada definición con su término mostrando todos en pantalla.
 - Probar: muestra término o definición y pregunta lo contrario. Pueden seleccionarse 3 tipos: escrita, opción múltiple y Verdadero/Falso. Así se pondrá a prueba al usuario.
- Clases: grupo en el que se comparten unidades con el resto de usuarios que pertenece a dicha clase.
- Carpetas: se pueden agregar unidades a las distintas carpetas creadas con el fin de organizar el contenido en relación al criterio del usuario.
- Incluye también un botón de búsqueda que permite buscar unidades, clases y usuarios.

El menú de ajustes dispone de las siguientes opciones:

- Perfil: muestra el usuario, las unidades creadas y las clases de las que es miembro.
- Comentarios: permite enviar comentarios a los desarrolladores de la aplicación.
- Valóranos: botón que enlaza con Play Store para valorar la aplicación.
- Política de privacidad: muestra la política de privacidad de la aplicación en un navegador externo.
- Cerrar sesión: cierra la sesión del usuario.

Esta aplicación cuenta con 5 millones de descargas en el Play Store de Android y un 4,2 sobre 5 puntos. Se han encontrado una serie de comentarios buenos y malos que se detallan a continuación.

2.1.4 Ventajas Quizlet

- Su interfaz es amigable que utiliza 2 colores (blanco y azul).
- Permite reutilizar los contenidos que han creado otros usuarios, así como pertenecer a un grupo en el que compartir contenidos.
- Se puede buscar a otro usuario para ver sus contenidos.
- Es totalmente completa desde la aplicación móvil.
- Permite respuestas abiertas al usuario.

2.1.5 Desventajas Quizlet

- No se pueden mezclar contenidos de varias unidades.
- No dispone de estadísticas ni guarda resultados sobre tests anteriores.
- Requiere atención del usuario para comprender como funciona cada funcionalidad. Los nombres no son intuitivos respecto a su función.

- En preguntas abiertas el usuario ha de contestar literalmente lo que tiene guardado en la base de datos la aplicación, no realiza un porcentaje de acierto.
- Solo permite introducir término + definición, no preguntas tipo test.

2.1.6 Conclusión del uso de Quizlet

Como conclusión, es una aplicación muy simple que solo permite evaluar sobre conocimientos de palabras. Acierta en la búsqueda de contenidos y usuarios. Su punto fuerte es ser autónoma en su versión móvil. La funcionalidad de pertenecer a una clase en la que poder colgar contenidos para otros usuarios es una funcionalidad muy potente ya que puede permitir a usuarios que estudian lo mismo compartir contenidos, sería una funcionalidad imitable. Mientras que limitarse exclusivamente a término + definición correcta es un punto débil que hay que evitar ya que si la aplicación permitiera término + definición correcta + definiciones incorrectas haría que los usuarios pudieran tener mejor experiencia al centrarse no solo en el conocimiento de definiciones de palabras.

2.1.7 Conclusión general sobre aplicaciones similares

Al haber estudiado en profundidad 3 de las aplicaciones más usadas por los usuarios (Una en versión web, una en versión móvil y otra con ambas versiones) se extraen las siguientes líneas que debe seguir la aplicación que se va a desarrollar para recopilar lo mejor de cada una y evitar lo peor. Siguiendo las conclusiones obtenidas y aplicando las heurísticas de Nielsen para crear un sistema amigable nos encontraríamos ante:

- **H2 Utilizar el lenguaje de los usuarios y H6 Minimizar la carga de la memoria del usuario:** los nombres de cada funcionalidad deben ser representativos con el recurso que ofrecen, es decir, debe favorecer la experiencia del usuario mediante iconos y nombres intuitivos.
- **H8 Diálogos estéticos y diseño minimalista:** se debe tener una interfaz amigable que no produzca cansancio al navegar por sus pantallas. Puede lograrse evitando al mínimo la publicidad y utilizando una paleta que no supere los 3 colores.

Además, como ideas que se han tomado con el fin de ofrecer al usuario una aplicación lo más completa posible, se han seleccionado las siguientes funcionalidades:

- Permitir puntuar y guardar los recursos de otros usuarios en el perfil de cualquier usuario.
- Incorporar un buscador de usuarios y contenidos por tema.
- Permitir preguntas abiertas extendiendo la funcionalidad de respuesta múltiple o Verdadero/Falso.
- Pestaña de mensajería para poder consultar/resolver dudas a otros usuarios sobre el contenido de los recursos.
- Asignar etiquetas a cada recurso para facilitar su búsqueda.
- Realizar y mostrar estadísticas de los resultados obtenidos en cada pregunta y test.

2.2 Sistemas multimodales

La Interacción Multimodal consiste en un proceso en el cual diversos dispositivos y personas son capaces de llevar a cabo una interacción (auditiva, visual, táctil y gestual) conjunta desde cualquier sitio, en cualquier momento, utilizando cualquier dispositivo y de forma accesible, incrementando así la interacción entre personas, y entre dispositivos y personas. [4]

El **objetivo** fundamental de un sistema multimodal es superar las limitaciones de la interacción basada exclusivamente en el habla. No se restringe a utilizar el habla como único canal de comunicación, sino que puede utilizar varios métodos de entrada como, por ejemplo teclado, ratón, micrófono, cámara, pantalla sensible al tacto, etc. Del mismo modo así podrán ser los métodos de salida no centrándose únicamente en uno.

En la aplicación que se está desarrollando, el sistema multimodal humano-dispositivo se centrará en 2 métodos de entrada-salida: pantalla táctil y reconocimiento oral, así el dispositivo se comunicará con la persona a través de gráficos y texto en la pantalla táctil y oralmente a través de los altavoces del dispositivo y el humano se comunicará a través de la pantalla táctil mediante escritura y a través de micrófono con la voz.

2.2.1 Funcionamiento

Debido al extenso crecimiento de los dispositivos de acceso a la información, arquitecturas, sistemas operativos, etc. el W3C (World Wide Web Consortium, en español Consorcio de WWW) trabaja en estándares y herramientas de integración de datos, para que no sea tan difícil que diferentes mecanismos puedan interactuar entre sí. A continuación se exponen los principales que reconoce W3C que serán los que se utilicen para explicar el funcionamiento a nivel de entradas y salidas de un sistema de diálogo multimodal [5].

- VoiceXML es un lenguaje de etiquetado que permite crear diálogos con los que se puede interactuar escuchando comandos hablados, controlables a través de entradas de voz. Se encarga de convertir habla en texto y para ello utiliza, entre otros mecanismos: SRGS.
- InkML es un lenguaje XML diseñado para facilitar el intercambio de escritura electrónica entre distintas aplicaciones. Captura movimientos, ángulo y presión de forma electrónica para su posterior almacenaje y reconocimiento. No sólo reconoce escritura (texto plano, matemáticas, química, etc.), sino que verifica firmas e interpreta gestos.
- SRGS (Gramática de reconocimiento del habla) permite que una aplicación de voz indique a un reconocedor, qué es lo que tiene que escuchar, es decir, palabras, modelos en los que estas palabras surgen, lenguaje hablado de cada palabra, etc.
- SSML (Lenguaje de Marcado de Síntesis del Habla) forma parte del proceso de salida de información y funciona como componente de conversión ayudando a generar habla sintética. Este lenguaje tiene como principal objetivo ofrecer a los autores de contenido sintetizable un camino estándar para controlar aspectos del habla como pueden ser la pronunciación, volumen, tono y velocidad, mejorando así la calidad del contenido sintetizado.
- SISR (Interpretación semántica del reconocimiento del habla) permite describir anotaciones sobre reglas gramaticales para extraer resultados semánticos a través del reconocimiento del habla. El resultado obtenido puede expresarse en formato XML, usando el lenguaje EMMA.
- EMMA (Extensible MultiModal Annotation) es un lenguaje utilizado para el intercambio de datos en sistemas de administración de interacción multimodal. Es una especie de lenguaje común utilizado en la comunicación entre componentes de un sistema multimodal, cuyo objetivo es integrar la entrada de datos procedente de los usuarios desde diferentes recursos, y darle forma para ser procesada en una representación única que será a su vez procesada por componentes avanzados de procesamiento de información.

En el capítulo 3, [sección 3](#) de este documento, se detalla cómo funcionan los sistemas multimodales: componentes de entrada y componentes de salida.

2.3 Android



Figura 6. Evolución de versiones Android

Android nace en 2003, siendo propiedad de Android Inc., de la mano de un ex trabajador de Apple y Microsoft. Cuando en 2005 esta compañía tenía 22 meses de antigüedad fue comprada por Google. Tras esto, sucedieron 2 años en los que no se supo nada de la evolución de esta empresa que había adquirido Google, cuando, a finales de 2007 se hace público el anuncio de Android como sistema operativo [6].

Hasta la actualidad, Android ha lanzado distintas versiones para sus dispositivos móviles que se detallan a continuación:

I. [Android 1.0: Apple Pie \(22 de octubre de 2008\)](#)

El primer Smartphone que funcionaba gracias a este SO fue el HTC Dream (Google Phone). Incluía el primer Market de Android, entre otras aplicaciones básicas de un teléfono móvil y junto a aplicaciones propiedad de Google como por ejemplo: Google Maps, Google Sync, Google Search, Google Talk y Youtube. Contaba además con un sistema de notificaciones que aparecían en la barra de estado.

II. [Android 1.1: Banana Bread \(9 de febrero de 2009\)](#)

Disponible solo para dispositivos T-Mobile G1 y cuya principal novedad fue el poder guardar los archivos adjuntos en los mensajes y las reseñas al buscar negocios en los mapas.

III. [Android 1.5: Cupcake \(30 de abril de 2009\)](#)

Fue la primera gran actualización de este SO, siendo los aspectos más llamativos la introducción del teclado virtual en pantalla y la posibilidad de insertar widgets. Además se podía hacer *Copy&Paste* en el navegador, grabar y reproducir formatos MPEG-4 y 3GP, permitiendo subir estos videos directamente a Youtube. Cabe destacar la auto rotación entre otras mejoras.

IV. [Android 1.6: Donut \(15 de septiembre de 2009\)](#)

Esta actualización corregía y mejoraba características ya existentes, como el cuadro de búsqueda, la cámara, la galería y el Android Market. Incluyó algo muy solicitado por los usuarios: poder tomar capturas de pantalla. Fue una actualización que no llegó a todos los dispositivos y muchos saltaron desde la 1.5 a la 2.1

V. Android 2.0: Eclair (26 de octubre de 2009)

El aspecto más importante que trajo consigo esta actualización fue la integración social pudiendo sincronizar los contactos de Facebook y Twitter. Además la interfaz pasó a tener mayor fluidez. Se mejoró notablemente la cámara dando al usuario funcionalidades que hasta entonces no disponía.

- Android 2.1 fue la segunda etapa en la evolución de Eclair con su introducción en el dispositivo Nexus One. Incorporaba mejoras a ésta como fondos interactivos, 2 nuevos escritorios (pasa de 2 a 5). Precisamente en Nexus One fue el primer teléfono en el que se extendían las **capacidades de voz** dando lugar a un micrófono en el teclado para complementar a lo que ya pudiera hacerse mediante la escritura.
- Android 2.1 (Multitouch) (12 de Enero de 2010) El *multitouch* que trajo consigo esta actualización fue una característica codiciada por los usuarios ya que hasta el momento solo los poseedores de un iPhone podían disfrutar.

VI. Android 2.2: Froyo (20 de mayo de 2010)

Fue la actualización que consagró a Android como SO frente a Apple (que en aquel momento contaba con iOS 4) ya que el incremento de velocidad en todo el sistema era más que notable. Este incremento de velocidad se produjo gracias al motor V8 de Java y el compilador JIT. Una de las grandes novedades fue la posibilidad de hacer **tretheing**, es decir, compartir conexión 3G del teléfono a través de redes *wifi* convirtiéndose en un *hotspot*. También cabe destacar la posibilidad de trasladar aplicaciones desde la memoria interna del dispositivo a una tarjeta *microSD*.

VII. Android 2.3: Gingerbread (6 de diciembre de 2010)

Con esta actualización, Android pasaba a ser más veloz y simple, y sobretodo, se preparaba para la llegada de los smartphones de doble núcleo. En esta actualización se incorporaron una gran gama de funciones que permiten manejar el dispositivo con la voz denominadas **Voice Actions**: enviar mensajes, realizar llamadas, localizar lugares vía GPS, realizar búsquedas, escuchar música, entre otras. Se incorporó también el sistema NFC.

VIII. Android 3.0: Honeycomb (22 de febrero de 2011)

Google comenzó a desdoblar el SO con esta última actualización más su correspondiente SDK, lo cual no tuvo mucho recorrido debido al coste de mantener dos plataformas independientes. Trajo consigo una nueva interfaz, se simplificó la función multitarea y se cambiaron las ventanas del navegador por pestañas. También es importante mencionar el soporte para procesadores con varios núcleos, aceleración del hardware, posibilidad de encriptación de los datos del usuario y mejora en el uso de HTTPS gracias a SNI.

IX. Android 4.0: Ice Cream Sandwich (19 de octubre de 2011)

Se renovó por completo la interfaz gracias al diseño Holo y se volvió a integrar el SO en sus versiones para Tablets y Smartphones. Los botones físicos desaparecieron (hardware) y llegaron los botones virtuales. Se incluyó el desbloqueo facial y una sección para controlar el consumo de datos pudiendo configurar límites. Esta actualización también mejoró en diversos aspectos la cámara y se incluía una aplicación para la edición de fotos.

X. Android 4.1: Jelly Bean (27 de Junio de 2012)

Actualización muy importante cuyo objetivo fue mejorar la estabilidad, funcionalidad y rendimiento de la interfaz de usuario. También se mejoró la barra de notificaciones, teniendo ésta una mayor integración gracias a poder realizar llamadas y otras acciones que antes no

podían realizarse desde ella. **Se incluyó Google Now, que junto a Knowledge Graph y su búsqueda por voz, consiguieron superar ampliamente a Siri.**

- Android 4.2: traía novedades como:
 - Photo Sphere que permitía tomar imágenes panorámicas en horizontal y vertical.
 - Gesture Typing: permite escribir deslizando el dedo sobre las letras.
 - Soporte para múltiples usuarios.
 - Quick Settings: mejora en la barra de notificaciones que permitía controlar distintas configuraciones del dispositivo.
- Android 4.3: introdujo mejoras de seguridad y rendimiento. Además incorporaba el soporte para perfiles restringidos.

XI. *Android 4.4: KitKat (31 de octubre de 2013)*

La mayor característica de esta actualización fue una reducción del tamaño del SO. Siguió utilizando la interfaz Holo pero modificó diversos aspectos como el tamaño de los iconos y la condensación del texto para una visión más clara y simple. Se mejoraba el rendimiento para dispositivos con una RAM de 512MB. En esta versión, se incluyó una máquina virtual, llamada ART, para desarrolladores.

XII. *Android 5.0: Lollipop (12 de Noviembre de 2014)*

Se renovó, una vez más, su interfaz con innovaciones y nuevas funcionalidades todo ello dentro de un nuevo diseño de Material Design, que buscaba colores llamativos, diseño intrépido y animaciones que simularan integración al usuario. Además de elementos estéticos se reforzó la batería una media de 90 minutos gracias a Project Volta. Las notificaciones siguen mejorando y se pueden controlar mediante franja horaria.

- Android 5.1: sus principales novedades fueron:
 - Soporte para múltiples tarjetas
 - Llamadas de voz en alta definición para usuarios con mismo SO.
 - Protección frente a robo o extravío.

XIII. *Android 6.0: Marshmallow (5 de octubre de 2015)*

Mejoras en rendimiento y estabilidad del SO. Destaca Google Now on Tap, mejoras en gestión de la batería gracias a Doze, soporte para huellas dactilares, USB tipo-C y Chrome funcionando dentro de otras aplicaciones. Permisos “opt-in” (las aplicaciones pedirán acceso antes de utilizar otras partes del SO).

XIV. *Android 7.0: Nougat (Actualidad)*

La última versión del SO introduce más de 250 prestaciones, convirtiéndolo así en la mejor versión de Android [7]:

- Nuevo entorno de ejecución de aplicaciones ART con compilación JIT: haciendo que la instalación sea más rápida.
- Project Svelte optimiza el consumo de RAM en aplicaciones en segundo plano.
- API gráfica Vulkan mejorando los gráficos con más efectos y fluidez entre ellos.
- Interfaz similar a la anterior.
- Pantalla dividida para poder usar dos aplicaciones a la vez.
- Agrupación de notificaciones y respuestas directas desde la barra de notificaciones.
- Permite personalizar la barra de ajustes.

Aspectos a destacar sobre las distintas versiones de Android

Como el tema de estudio de este TFG son las interfaces orales, sería conveniente mencionar a modo de resumen que a lo largo de la historia de las distintas versiones de Android que se acaban de enunciar brevemente cabría destacar que:

- En la versión 2.1 se extendieron las capacidades de voz mediante la introducción de un micrófono en el teclado para complementar a la escritura.
- En la versión 2.3 se introduce la funcionalidad Voice Actions para poder enviar mensajes, realizar llamadas, localizar lugares vía GPS, realizar búsquedas, escuchar música, entre otras cosas.
- En la versión 4.1 se incluyó Google Now que junto a Knowledge Graph y su búsqueda por voz, consiguieron superar ampliamente a Siri.

2.3.1 Android Studio

Android Studio nació el 16 de Mayo de 2013 siendo anunciado en la conferencia de Google I/O, aunque su primera versión estable no estuvo disponible hasta diciembre de 2014. Las plataformas en las que puede ser usado son las 3 más grandes del mercado: Microsoft Windows, Mac OS y Ubuntu Linux [8].

Gracias a este programa, Google se beneficia teniendo su propio creado de aplicaciones para Android, para el usuario las ventajas vienen de la mano de Google también, ofreciendo así un software con actualizaciones continuas. Es uno de los IDE más completo para desarrollar en Android. Facilita el trabajo de desarrolladores, teniendo estos un fácil acceso a sus carpetas, archivos y demás documentación que se esté usando para el desarrollo. Se puede descargar gratuitamente desde su página oficial en el siguiente [enlace](#)

En el caso de este proyecto se va a realizar sobre Windows, por lo que las características mínimas son las siguientes [9]:

- Windows 7/8/10 (32 o 64 bits).
- 2 GB de RAM (8 GB de RAM recomendado).
- 2 GB de espacio libre mínimo (4 GB recomendado).
- Resolución mínima de 1.280 x 800.
- Java 8.
- 64 bits y procesador Intel (emulador).

La decisión de seleccionar Android Studio sobre Eclipse con el plugin de Android fue un compendio entre la recomendación del tutor del TFG y la motivación del alumno por aprender a usar una herramienta nueva no vista en la universidad. Además de esto, tras buscar por internet cuáles serían las ventajas de utilizarlo, sería conveniente destacar; las continuas actualizaciones; la programación se realiza sobre un programa exclusivo para Android y no sobre un programa que se nutre de un plugin; el código se muestra más ordenado y estructurado; y la no necesidad de descargar un *builder* para diseñar las interfaces [10].

2.3.2 Proyecto en Android Studio

Al crear un nuevo proyecto se deberá especificar el nombre de la aplicación, el dominio y la ruta donde se creará el proyecto. Se deberán seleccionar las plataformas y las APIs que se utilizarán. Ya que la aplicación será para dispositivos móviles, se seleccionarán teléfonos y tablets y ya que se quiere que llegue al público más amplio posible, se seleccionara una API de versión reducida (La versión mínima que se seleccione implica los dispositivos que podrán

descargarla). Ya se han seleccionado los valores iniciales, ahora antes de poder comenzar con el desarrollo, se ha de seleccionar la actividad principal, en este programa se refiere a la pantalla inicial que mostrará la aplicación. Para poder diseñar la interfaz principal al gusto del desarrollador, se selecciona *EmptyActivity* que mostrará una pantalla vacía. En este momento definiremos su clase java asociada (lógica de negocio) y su *layout xml* (interfaz gráfica). Una vez se tiene creado el proyecto, se va a comentar la estructura del mismo [11]:

- Dentro de un proyecto pueden existir varias aplicaciones (módulos) que se observan en la cada carpeta app dentro del proyecto. En el caso de este desarrollo, solo se tendrá un módulo.
- Dentro de app, se observan 3 subcarpetas: *androidTest*, *main* y *test*. *Main* es la carpeta en la que se desarrollan las interfaces y la lógica de negocio, por tanto será en la que se profundice sobre la estructura del proyecto.
 - Java: incluirá el código fuente de la aplicación.
 - Res: incluirá los recursos necesarios para el proyecto:
 - *Layout*: almacenará las distintas interfaces XML. Podrán crearse en vista de diseño o texto plano.
 - *Midmap*: almacenará los iconos de la aplicación. En este caso será en *midmap-hdpi* por la densidad de la pantalla.
 - *Values*: almacenará estilos, colores, etc.
 - *AndroidManifest.xml*: contiene la definición de aspectos principales de la aplicación: nombre, icono, componentes, permisos, etc.
- En el fichero *build.gradle* se almacena la información necesaria para la compilación del proyecto: versión de SDK, versión mínima de Android que soportará la app, referencias a librerías, etc.
- En la carpeta *libs* se guardarán los ficheros *jar* (librerías externas) que utilice la aplicación. Pueden guardarse en esta carpeta o en el fichero mencionado en el punto anterior.
- En la carpeta *build* se generan elementos de código necesarios para la compilación. No deben ser tocados manualmente.

Estas son las principales carpetas y ficheros necesarios en Android Studio para desarrollar una aplicación.

2.4 Bases de datos

“Una base de datos es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.” [12]

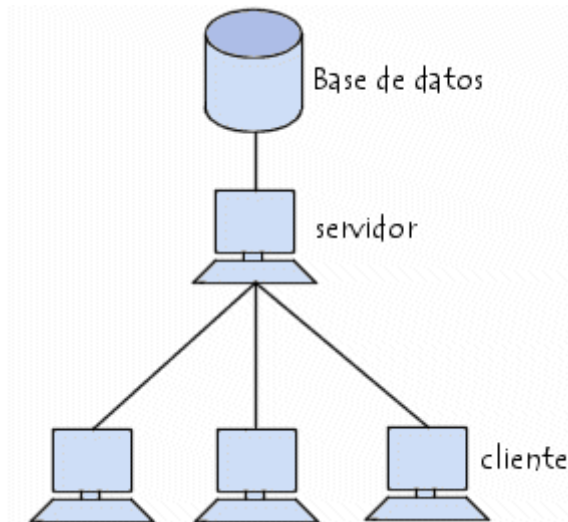


Figura 7. Base de Datos distribuida

Una base de datos permite a sus usuarios visualizar, insertar, borrar o actualizar la información que contiene.

Las bases de datos pueden ser; locales, solo se puede acceder a la información desde el equipo en que está alojada; o distribuidas, la información se almacena en equipos remotos y se accede a ella a través de la red.

2.4.1 SGBD – Sistema de Administración de Base de Datos

Para poder controlar los datos que contiene y los usuarios que tienen acceso a ella es necesario disponer de un Sistema Gestor de Base de Datos (en adelante SGBD), también conocido como DBMS por sus siglas en inglés (DataBase Management System). Este sistema permite a múltiples usuarios fácil acceso y manipulación de los datos.

La arquitectura de este sistema especifica sus componentes, que son los siguientes [13]:

- Interfaces externas: medios para comunicarse con el SGDB en ambos sentidos (E/S) y explotar a todas sus funciones. Pueden afectar a la BBDD o a la operación del SGBD.
- Intérprete o procesador del lenguaje: la mayor parte de las operaciones se efectúan mediante un lenguaje de base de datos. Existen lenguajes para definición de datos, manipulación de datos (p. e. SQL), para especificar aspectos de la seguridad y más. Las sentencias en ese lenguaje se introducen en el SGBD mediante la interfaz adecuada. Se procesan las expresiones en dicho lenguaje (ya sea compilado o interpretado) para extraer las operaciones de modo que puedan ser ejecutadas por el SGBD.

- Optimizador de consultas: realiza la optimización de cada consulta y escoge el plan de actuación más eficiente para ejecutarlo.
- Motor de la base de datos: realiza las operaciones requeridas sobre la base de datos, típicamente representándolo a alto nivel.
- Mecanismo de almacenamiento: traduce las operaciones a lenguaje de bajo nivel para acceder a los datos. En algunas arquitecturas el mecanismo de almacenamiento está integrado en el motor de la base de datos.
- Motor de transacciones: para conseguir corrección y fiabilidad, la mayoría de las operaciones internas del SGBD, se realizan encapsuladas dentro de transacciones. Las transacciones pueden ser especificadas externamente al SGBD para encapsular un grupo de operaciones. El motor de transacciones sigue la ejecución de las transacciones y gestiona su ejecución de acuerdo con las reglas que tiene establecidas (p. e., control de concurrencia y su ejecución o cancelación).
- Gestión y operación de SGBD: comprende muchos otros componentes que tratan de aspectos de gestión y operativos del SGBD como monitoreo de prestaciones, gestión del almacenamiento, mapas de almacenamiento.

Los 5 SGBD más usados actualmente son los siguientes; MySQL; Microsoft SQL Server; Oracle; Microsoft Access; PostgreSQL. Todos ellos son relacionales (un ejemplo sencillo para entender como es una base de datos relacional es imaginarse una tabla con registros “tuplas” y columnas “campos”) y dos son de software libre para usuarios, es decir, pueden usarse sin tener que obtener una licencia de pago, estos son MySQL y PostgreSQL [14].

El más utilizado en el ámbito de la enseñanza es MySQL ya que, además de ser gratuito para usuarios, su configuración e instalación es fácil, no supone gran coste en requerimientos para la elaboración de la BBDD y es veloz al realizar operaciones. Ya que es el SGBD que se ha aprendido en la universidad y no forma parte del objetivo de aprendizaje del TFG sino como complemento para poder dotar a la aplicación de una base de datos en la que almacenar información, será el sistema escogido para ello por lo que se profundizará en las siguientes líneas en sus principales características.

2.4.2 MySQL

Como ya se comentó en el punto anterior, es un SGBD bajo licencia libre *GNU* para usuarios (siendo la más popular del mundo) y licencia comercial *Enterprise* para empresas. Lo desarrolla *Oracle Corporation*.

Se puede acceder a las bases de datos MySQL desde diversos lenguajes como *C*, *C++*, *C#*, *Pascal*, *Delphi*, *Eiffel*, *Smalltalk*, *Lisp*, *Perl*, *PHP*, *Python*, *Ruby*, *Gambas*, *REALbasic*, *x(Harbour)*, *FreeBASIC*, *Tcl* y *Java*. Este último es el más interesante para este proyecto ya que la parte de lógica de negocio y persistencia se programará en Java para los métodos que interaccionan entre sí y las consultas/inserciones/actualizaciones frente a la base de datos.

Actualmente, la versión estable de este programa es la 6.3.7 y puede descargarse desde la página oficial de [MySQL](https://www.mysql.com/).

Las plataformas para las que está disponible son: *Microsoft Windows, Ubuntu Linux, Red Hat Enterprise Linux/Oracle Linux, Fedora, Mac OS*. Además se puede descargar el código fuente para implementarlo en un sistema que no sea alguno de los citados anteriormente.

A estas características de multilenguaje y multiplataforma, habría que añadirle características exclusivas que diferencian MySQL de otro SGBD:

- Permite escoger entre múltiples motores de almacenamiento para cada tabla en tiempo de ejecución. Algunos ejemplos de motores son *MySAM, Falcon, Merge, InnoDB, MySQL, etc.*
- Agrupación de transacciones para incrementar el número por segundo.

También sería conveniente mencionar que ofrece una conectividad segura, la posibilidad de seleccionar mecanismos de almacenamiento con el fin de mejorar la velocidad de operación, el soporte físico, la capacidad, distribución geográfica, y transacciones, entre otras.

Por la licencia libre que ofrece, *Java* como lenguaje de acceso, *Microsoft Windows* como Sistema Operativo en el que descargar el instalador y las características mencionadas, sumado al conocimiento del programa *MySQL* cumple los requisitos para ser utilizada como base de datos en el desarrollo de la aplicación para guardar la información que más espacio de almacenamiento necesita.

CAPÍTULO 3. DESCRIPCIÓN GENERAL DE LA APLICACIÓN DESARROLLADA

Tras haber estudiado y profundizado en el marco en el que se va a desarrollar la aplicación, es decir, en los puntos tratados dentro del estado del arte: aplicaciones similares que se encuentran en el mercado, sistemas de diálogo multimodales actuales y su funcionamiento, Android Studio como herramienta principal sobre a qué se va a trabajar y las distintas bases de datos más usadas y cuál ha sido seleccionada para guardar los datos de la aplicación. Se procede a describir detalladamente a lo largo de este punto:

- Presentación de la aplicación: funcionalidades que posee.
- La arquitectura seguida en el desarrollo (MVC).
- Interfaz oral: componentes de entrada y salida y ejemplos.

Inicialmente se explicarán las líneas principales que sigue la aplicación desde las interfaces que se han diseñado para que el usuario pueda navegar por las distintas pantallas de la aplicación, explicando la capa de lógica de negocio a través de ellas y llegando al esquema relacional para explicar cómo se almacena la información en la base de datos seleccionada. Se finalizará explicando cómo funciona una interfaz oral en un dispositivo móvil.

3.1 Presentación de la aplicación

Esta aplicación permite a sus usuarios poner a prueba de los conocimientos que están estudiando mediante preguntas que previamente ellos mismos han insertado en la aplicación. El usuario tiene la posibilidad de insertar y resolver preguntas mediante dos interfaces: una táctil y otra oral.

La aplicación permite al usuario:

- Insertar/Actualizar/Borrar preguntas de la aplicación.
- Autoevaluarse con preguntas que previamente creó.
- Consultar estadísticas acierto/error y ver su progresión.

3.1.1 Inicio de sesión/Registro

Al abrir la aplicación se visualiza una pantalla en la que el usuario puede introducir sus credenciales en el caso de tener una cuenta creada previamente o registrarse para poder acceder a la aplicación. Si el usuario tiene una cuenta, deberá introducir su correo electrónico y contraseña y pulsar sobre *INICIAR SESIÓN*. Si no tiene cuenta y pulsa sobre *REGISTRARSE*, se mostrará una pantalla que requiere datos para crear una cuenta.



Figura 8. Vista de la aplicación: Pantalla de inicio



Figura 9. Vista de la aplicación: Pantalla de registro

- *INICIAR SESIÓN*: se realiza una consulta sobre la base de datos para comprobar si el e-mail del usuario existe en la base de datos y si la contraseña introducida coincide con la guardada.
- *REGISTRARSE* → *Pantalla de Registro* → *REGISTRO*: se realiza una inserción en la base de datos. Para ello, los datos recogidos por la interfaz de usuario se envían desde la capa de lógica de negocio a la de persistencia para que se realice una conexión *Android Studio* – *MySQL* y se realice la inserción de un usuario.

Tras completarse el proceso de registro o iniciar sesión, el usuario podrá acceder a su perfil y utilizar las funcionalidades que ofrece la aplicación y que recoge el menú principal. Introduciendo una asignatura y un tema en los 2 campos superiores, el usuario podrá realizar un test, insertar una pregunta, ver las preguntas ya creadas o ver su progresión. En caso de no existir la asignatura y el tema deseados, se crearán. Para ello, al introducir una asignatura y tema, se realizará una consulta contra la base de datos y de no existir en ésta, se hará una inserción sobre ella.



Figura 10. Vista de la aplicación: Pantalla de selección

Como se observa en estas 3 pantallas, la interfaz sigue los mismos colores. Se han predefinido como constantes dentro del código XML, y distribución con el fin de crear un entorno amigable con el usuario. Como se especificó en el estado del arte gracias al estudio de aplicaciones similares y a las heurísticas de Nielsen, se busca una pantalla que no produzca rechazo debido al uso de contenido flash o colores muy llamativos o distintos entre sí. Se evitan pantallas que no sigan dicho patrón para no hacer sentir al usuario perdido. Siempre que el usuario clique encima del logo, que se encuentra situado en el centro de la parte superior de la pantalla, la aplicación lo dirigirá a la pantalla de selección.

En *Android Studio*, como se detalló en el estado del arte, las interfaces se realizan en XML mediante *layouts*. Un *layout* es un contenedor que recoge las diversas *views* (las vistas, por ejemplo, un botón). En ese contenedor se define la estructura y el orden de esas vistas. Para la definición de estas interfaces se ha escogido uno de los *layouts* más usados *RelativeLayout* que permite organizar los componentes de forma relativa entre sí, así se podrán adaptar fácilmente a distintos tamaños de pantalla.

3.1.2 Insertar y visualizar preguntas

El usuario tiene la capacidad de decisión entre las opciones que se encuentran en la pantalla de selección. Al clicar sobre *INSERTAR PREGUNTA* se mostrará una pantalla en la que podrá, con el teclado gracias a la sensibilidad táctil de la pantalla o con su voz gracias a la interfaz oral, insertar una preguntas y respuestas en los cuadros destinados a ello, siempre teniendo un texto descriptivo que le guíe para no sentirse perdido.

Además existen 6 recuadros adicionales en los que podrá incluir palabras clave. Estas palabras clave extienden la capacidad de una aplicación meramente de tests proporcionando al usuario la opción de escribir o dictar un texto a su dispositivo en lugar de escoger una respuesta previamente creada. Esta funcionalidad se explica a lo largo de este sub capítulo.



Figura 11. Vista de la aplicación: Inserción pregunta

Como se ha explicado, esta pantalla permite al usuario insertar una pregunta, una respuesta correcta, tres respuestas incorrectas y 6 palabras clave. Al hacer clic sobre *GUARDAR PREGUNTA* la aplicación realizará un INSERT en la base de datos que guardará toda la información que se haya escrito en cada recuadro para posteriormente poder hacer consultas, actualizaciones o borrados sobre ellas.

Si se desea hacer uso de esta opción para editar preguntas insertadas, se deberá dirigir a la pantalla principal y en ella se ha de pulsar sobre *VER PREGUNTAS*. Se traslada al usuario a otra pantalla nueva en la que se muestran las preguntas guardadas en la base de datos con un límite de 6 por pantalla, pudiendo desplazarse entre estas gracias a los botones inferiores < y >



Figura 12. Vista de la aplicación: Visualización preguntas

Estas preguntas aparecerán en pantalla gracias a un bucle que recorrerá la tabla que contiene las preguntas. Se realizarán 6 consultas cada vez que se acceda a esta opción o el usuario decida pasar a la siguiente visualización, recuperando únicamente el campo mostrado, es decir, pregunta.

Al clicar sobre una de ellas, se accederá a la pantalla de Inserción de preguntas con la diferencia de que los campos rellenos. Se podrá editar o eliminar la pregunta mediante el comando *UPDATE* o *DELETE* en la base de datos.

3.1.3 Realizar un TEST

Cuando el usuario ya ha introducido en su perfil preguntas está preparado para poder poner a prueba sus conocimientos.

Para ello, deberá clicar sobre la opción *TEST* entre las posibilidades de selección. Una vez dentro, el usuario vuelve a tener que tomar una decisión y es que, a diferencia de otras aplicaciones similares como las estudiadas, la innovación que se incluye es poder realizar una pregunta abierta en la que no se tiene que contestar una de las respuestas cargadas con anterioridad ni se debe escribir exactamente lo que se puso al crearlo (lo cuál sería complicado pero una de las aplicaciones estudiadas en el estado del arte ofrecía a sus usuarios). La funcionalidad añadida es la que se encuentra tras el botón *PREGUNTA ABIERTA*.



Figura 13. Vista de la aplicación: Selección tipo TEST

- **PREGUNTAS TEST:** el usuario observa la pregunta inicial de las 10 que se sucederán. En cada una, se debe seleccionar una respuesta entre las opciones, como los test de siempre. Al querer pasar a la siguiente pregunta, se muestra la opción correcta de la actual. Al finalizar obtiene un resultado que evalúa cuantas preguntas ha contestado correctamente y cuantas incorrectamente. Estos datos se incluyen en cada pregunta guardada en la base de datos para poder mostrar estadísticas y progresión al usuario. Las 10 preguntas se seleccionan de la base de datos con un bucle existente en la base de datos en el que existe un método aleatorio para obtener 10 números que serán los identificadores de las preguntas.

- **PREGUNTA ABIERTA:** se muestra una pantalla con una de las preguntas cargadas en la base de datos (gracias al método aleatorio mencionado anteriormente) y un recuadro blanco sobre el que deberá contestar mediante el teclado o con su voz. Se copiará el texto y comparará este con las palabras clave que guardó al insertar la pregunta en la base de datos, contrastando si ha dicho esas palabras y en la medida de que lo haya realizado o no, otorgará un porcentaje que se visualizará al finalizar su respuesta. Para su comprobación se ha utilizado un diccionario que elimina preposiciones y solo se queda con sustantivos. Tras la poda, se contrasta cada uno de los sustantivos con las palabras clave y se le otorga una puntuación que va desde 0 a 100 en tramos de 16,67 puntos, es decir, cada palabra acertada sumará esa cantidad a la nota final.

3.1.4 Estadísticas y progresión del usuario

Una vez el usuario ha navegado por la aplicación realizando inserciones y contestando a preguntas, comenzará a llenarse la tabla que se observa en la siguiente figura:



Figura 14. Vista de la aplicación: Estadísticas y progresión del usuario

La pantalla se divide en 2 secciones *PREGUNTAS TEST* y *PREGUNTAS ABIERTAS*. Ambas muestran el número de; preguntas contestadas; aciertos; y fallos. Además la última fila de tabla muestra el porcentaje de acierto.

Para implementar esta pantalla, se recuperan datos de la base de datos. Se realiza un sumatorio de las columnas *contador_test*, *contador_abierta*, *acierto_test*, *fallo_test*, *acierto_abierta* y *fallo_abierta* de la tabla pregunta. En preguntas abiertas se considera fallo al superar el 50% del porcentaje, y fallo al no superarlo.

Para mostrar el porcentaje de acierto, se divide el número de aciertos entre el número de preguntas contestadas. Dichos resultados se operan en la lógica de negocio tras haberse recuperado gracias a la capa de persistencia.

3.2 Arquitectura de la aplicación

Para el desarrollo de la aplicación se ha realizado una implementación en 3 capas que se identifica con el famoso patrón de arquitectura *MVC* (Modelo Vista Controlador). En este patrón se realiza la organización independiente del modelo (datos), vista (interfaz) y el controlador (lógica). Dividiendo el sistema en capas, se encapsulando los datos, la interfaz visual y la lógica de forma que ninguna de las capas conoce el interior de las otras [15].

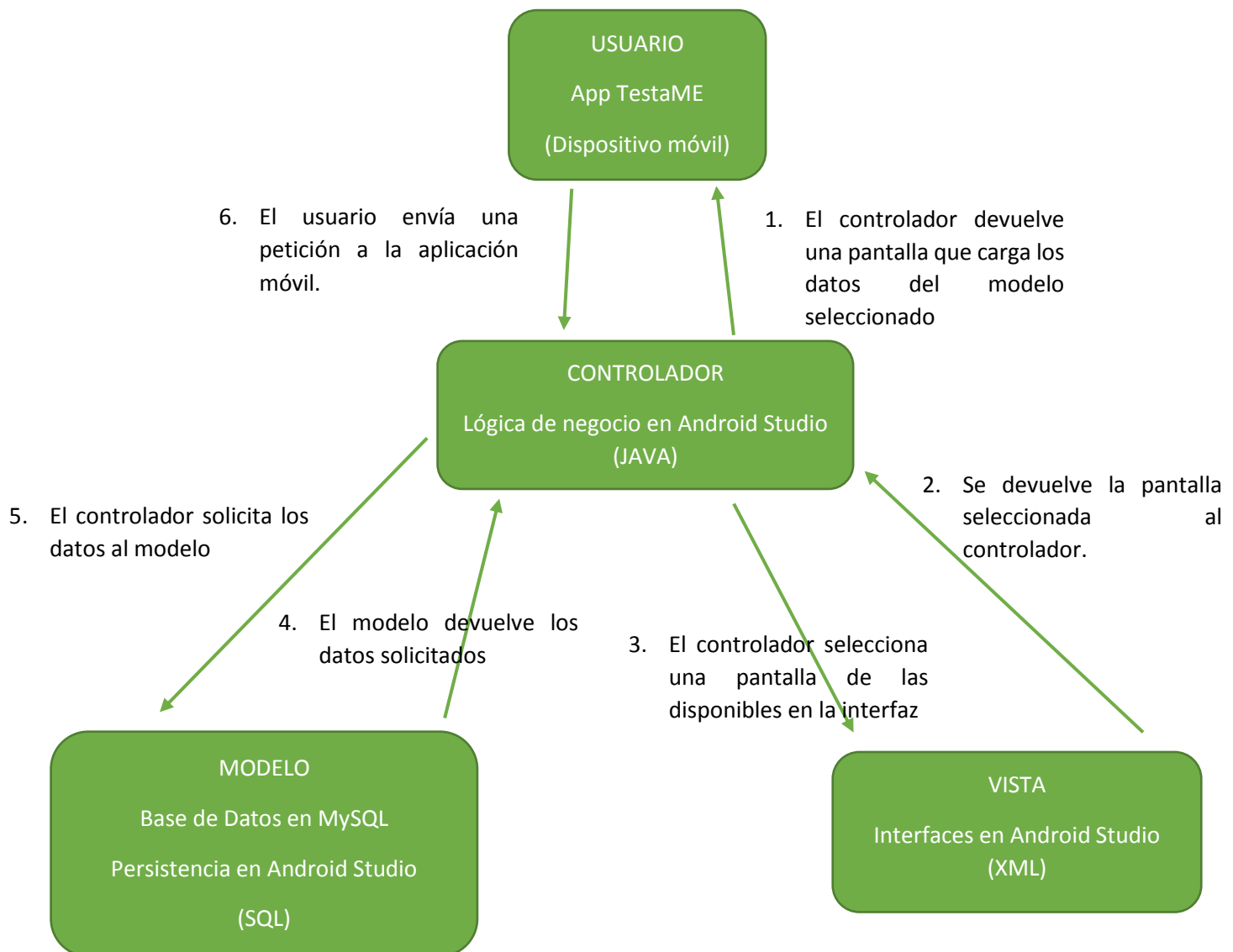


Figura 15. Arquitectura MVC

3.2.1 Modelo

En esta capa se ubican los datos de la aplicación que persisten entre sesiones. Se encapsula el estado de la aplicación y es independiente del controlador y la vista, existe como entidad propia sin necesidad de las otras dos capas.

Para la implementación del modelo, se han utilizado dos aplicaciones: *Android Studio* y *MySQL*.

En primer lugar, se explica el diagrama relacional [20] de tablas desarrollado en *MySQL*.

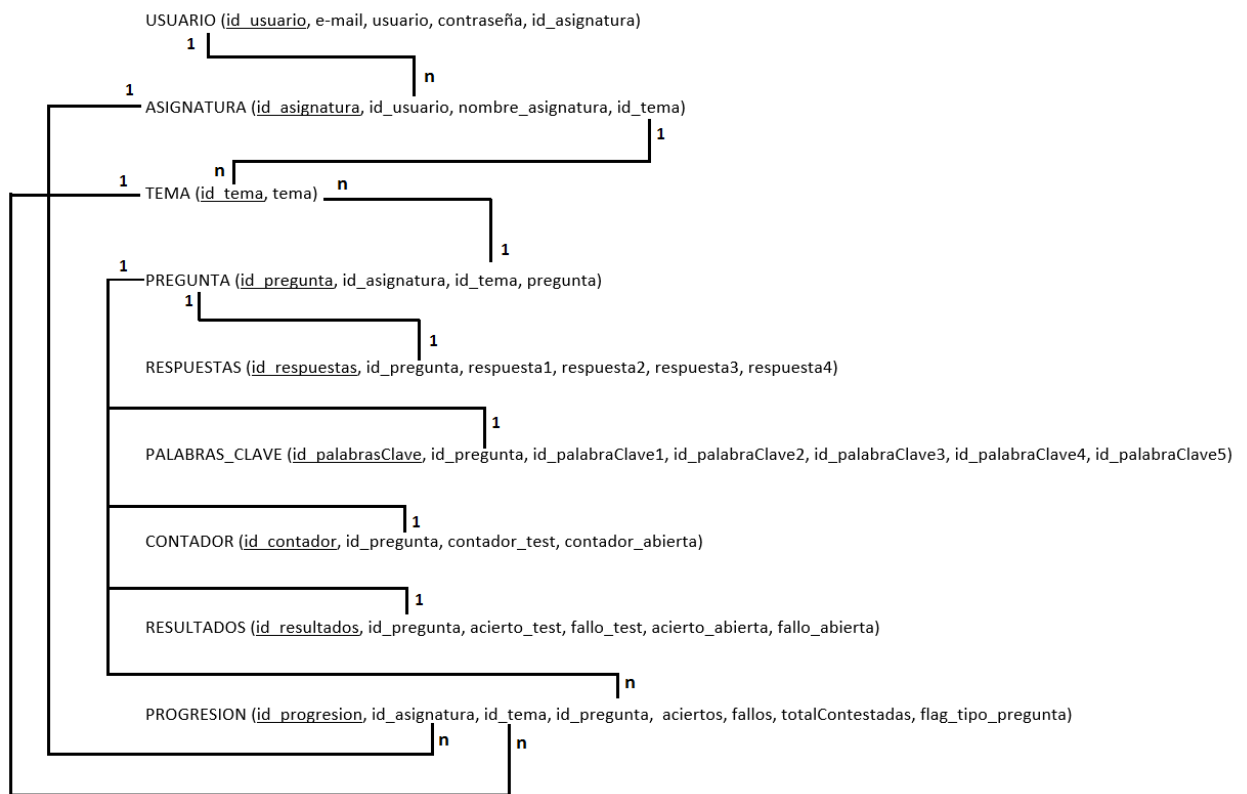


Figura 16. Esquema relacional BBDD aplicación

Como se observa en la figura, la base de datos contiene 9 tablas que están interconectadas entre sí para guardar información de usuarios, asignaturas, temas, preguntas, respuestas, palabras clave, contador de preguntas, resultados de preguntas y progresión.

- Gracias a estas tablas se pueden guardar datos persistentes como:
- Creación de usuarios
- Inserción/modificación/borrado de preguntas
- Actualización de resultados

Estos datos al ser persistentes, permanecen al finalizar las sesiones de la aplicación y son consultados mediante la implementación en java de *queries* SQL.

3.2.2 Vista

Esta capa es necesaria para mostrar al usuario de una forma ordenada los datos contenidos en el modelo. Cómo se explicó en el punto de presentación de la aplicación se compone de pantallas sobre las que el usuario navega.

Se ha desarrollado gracias al uso de etiquetas *XML* y *layouts* en la aplicación Android Studio.

A continuación se muestra una figura con los nombres de las distintas pantallas en el código fuente de la aplicación:

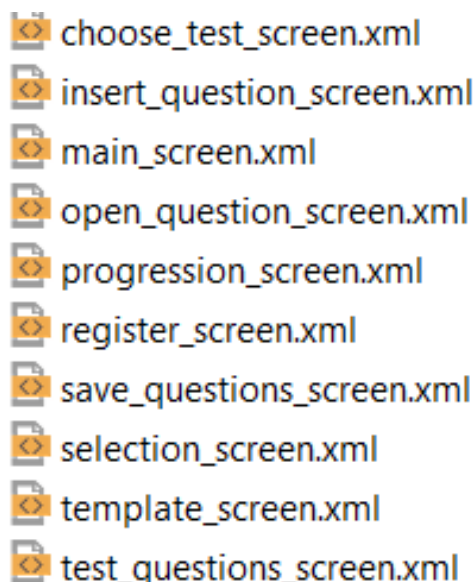


Figura 17. Lista de pantallas en Android Studio

3.2.3 Controlador

Esta capa es núcleo entre las otras dos, recoge los *clicks* realizados en la interfaz (vista) y es la encargada de enviar a la capa de persistencia (modelo) las consultas que se realizan para recuperar los datos desde esta última capa.

Además es la capa encargada de realizar cálculos con los datos recuperados y modificarlos en la capa de persistencia.

Se detalla a continuación como se produce la comunicación entre capas:

- Para su conexión con la capa de presentación (vista) se vinculan los botones, imágenes, campos de texto, etc. implementados en XML con el controlador gracias a lenguaje java.
- Para su conexión con la capa de persistencia (modelo) se crea una conexión con el *driver JDBC*. Para ello se debe descargar la librería necesaria desde la página web de *MySQL* y añadirla al proyecto de *Android Studio*. La conexión entonces se establece con abriéndola gracias a los siguientes campos: usuario, contraseña, nombre de la base de datos y *url* (ésta será una *url* local).

3.3 La interfaz oral

La aplicación permite al usuario la inserción de texto en los campos destinados a ello mediante una interfaz oral. Esta interfaz oral se activa gracias al uso de la interfaz táctil, donde tras situarse en un campo de texto y pulsar sobre él, aparece el teclado y entre sus opciones se puede observar un micro como se ve a continuación (marcado en rojo):

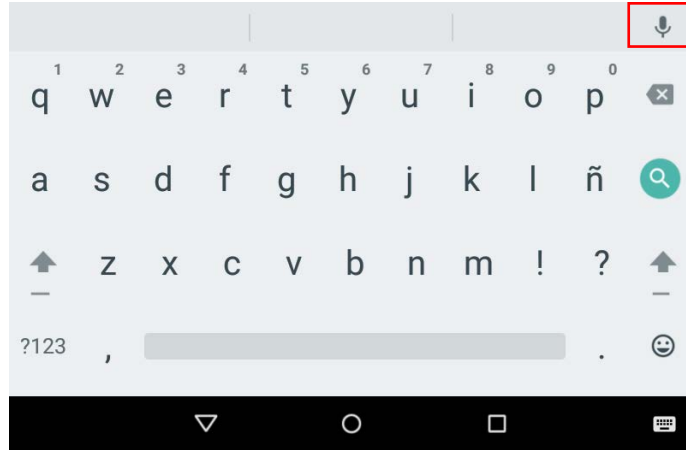


Ilustración 1. Interfaz oral: vista de micro en el teclado

Al finalizar la comunicación, se debe repetir el proceso para pasar al siguiente cambio o guardar los contenidos.

Para la transcripción se utilizan los paquetes incluidos en *android.speech*.

En los siguientes apartados se explica cómo funciona el reconocimiento multimodal, en este caso destacaría el reconocimiento de voz.

3.3.1 Componentes de entrada

La siguiente figura ha sido extraída del sitio web citado en el punto [5] de las referencias.

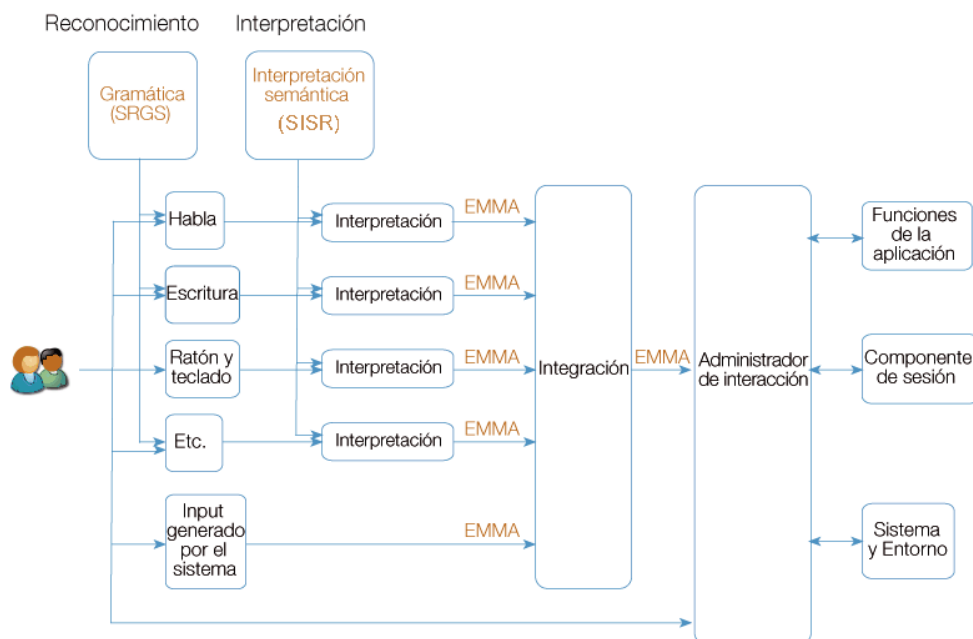


Figura 18. Componentes de entrada en un sistema de diálogo multimodal

Los componentes de reconocimiento recogen la entrada de datos del usuario mediante y la transforman para su procesamiento. Un componente de reconocimiento en la aplicación que se está desarrollando incluirá:

- Habla (VoiceXML): convierte el habla en texto. A través del micrófono escuchará sonidos y a través de un modelo acústico del lenguaje y una gramática específica usando SRGS convierte la voz humana en palabras concretas.
- Escritura (InkML): convierte en texto símbolos escritos y mensajes. El componente de reconocimiento de escritura utiliza un modelo de lenguaje y una gramática para convertir la escritura en palabras específicas.
- Teclado: convierte el uso del teclado en caracteres de texto.

Los componentes de interpretación se encargan de procesar los resultados que previamente se han transformado. Estos componentes identifican el significado o semántica deseada por el usuario gracias a SISR.

Los componentes de integración se encargan de combinar la salida de datos desde distintos componentes de interpretación.

3.3.2 Componentes de salida

La siguiente figura ha sido extraída del sitio web citado en el punto [5] de las referencias.

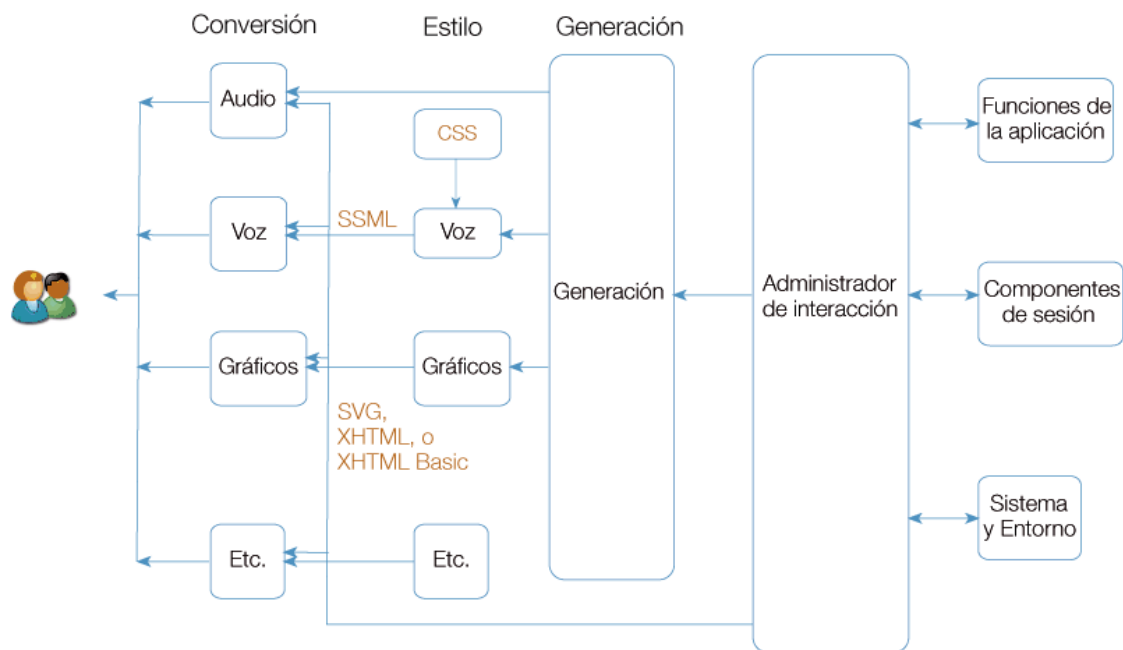


Figura 19. Componentes de salida en un sistema de diálogo multimodal

Los componentes de generación son los modos de salida que se van a utilizar para presentar la información desde el administrador de interacción hasta el usuario.

Los componentes de estilo se encargan de añadir datos sobre cómo se va a mostrar la información.

Los componentes de conversión se encargan de transformar la información del componente de estilo en un formato que será de fácil comprensión para el usuario.

3.3.3 Ejemplo input-output aplicación a desarrollar

En el siguiente diálogo se puede observar cómo sería una interacción entre el usuario de la aplicación y su dispositivo móvil.

Usuario: Pulsa sobre el recuadro para introducir una pregunta y dice “¿En qué año se descubrió América?”

Componente de reconocimiento del habla: Reconoce las palabras “¿En qué año se descubrió América?”

Componente de interpretación del habla: Convierte las palabras “¿En qué año se descubrió América?” en una notación interna.

Componente del administrador de la interacción: Almacena la notación interna en el objeto de la sesión. Convierte la petición en una petición de una base de datos, envía la petición a un sistema de administración de peticiones, la cual hará insertará dicha pregunta a la base de datos. Añade la respuesta a la notación interna en el objeto de sesión.

Componente de generación: Accede al componente de entorno para determinar que los modelos de voz gráficos están disponibles. Decide presentar el resultado como un modelo, texto en pantalla. El componente de generación envía una notación interna representando la frase escrita “¿En qué año se descubrió América?” al componente de estilo de escritura.

Componente de estilo de escritura: Convierte en SSML la notación interna que representa “¿En qué año se descubrió América?”

CAPÍTULO 4. EVALUACIÓN PRELIMINAR DE LA APLICACIÓN

Durante el desarrollo y una vez terminado este, se ha sometido a la aplicación a una batería de pruebas con el fin de testar la aplicación antes de su lanzamiento al mercado. Se deseaba depurar la aplicación para que el usuario que la descargase no encontrará limitaciones o utilidades que no se podían llevar a cabo. Además, se ha realizado una encuesta a usuarios potenciales de la aplicación para comprobar si las funcionalidades cumplen sus deseos o no, y en ese caso mejorar en futuras versiones de la aplicación.

4.1 Entorno de pruebas

En el entorno de pruebas se diferenciará entre el emulador en el que se testearán los resultados en el ordenador sobre la herramienta utilizada para el desarrollo y por otro lado un dispositivo móvil.

4.1.1 Emulador Android Studio

Para realizar las pruebas sobre *Android Studio*, el emulador se creó con las siguientes características:

- Galaxy Nexus de 4,65" con una resolución de 1280x720 y una densidad de *xhdpi*

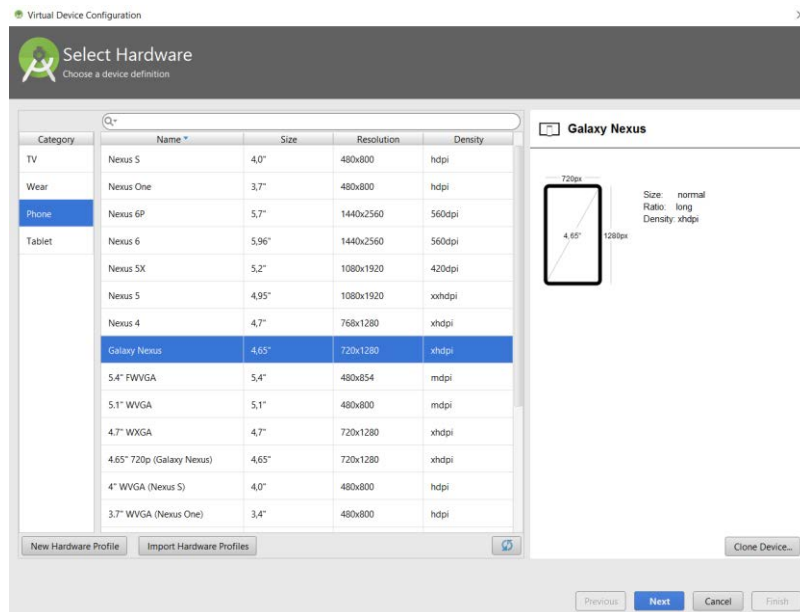


Figura 20. Selección hardware emulador Android Studio

- Android Lollipop (5.0) → API 22

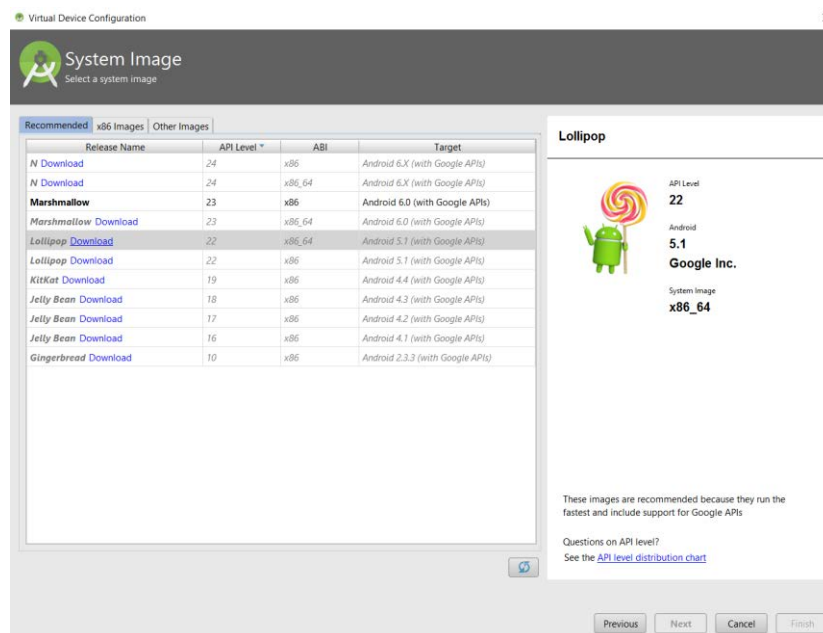


Figura 21. Selección software emulador Android Studio

A modo de resumen del emulador creado: el tamaño de pantalla es de 5" y versión de Android es Lollipop. Se ha seleccionado una escala automática y una orientación de la pantalla vertical, ya que la aplicación en su versión inicial solo podrá ser usada con el dispositivo en vertical para facilitar la experiencia al usuario. Estas indicaciones se han seleccionado para que el emulador de la herramienta *Android Studio* sea lo más parecido posible al dispositivo móvil físico disponible para la realización de las pruebas, el cuál se detalla en el siguiente punto.

4.1.2 Dispositivo móvil

Para realizar las pruebas en un dispositivo externo a la herramienta utilizada para codificar la aplicación se ha utilizado un Smartphone LG G3S. Sus características se detallan a continuación:

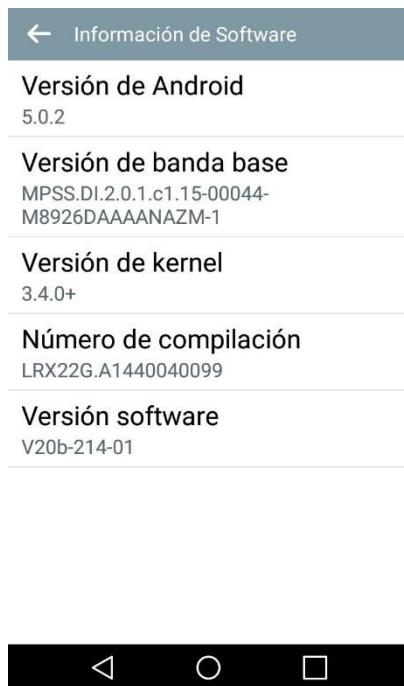


Figura 22. Información del Software del dispositivo móvil



Figura 23. Versión de Android del dispositivo móvil

Además de estas características técnicas, la pantalla es de 5" y tiene una resolución de 1280x720 (HD IPS)

4.2 Pruebas realizadas

En este punto se detallan las pruebas a las que se ha sometido a la aplicación para comprobar que funciona correctamente.

Para la especificación de las pruebas se ha elaborado la siguiente tabla:

ID	
Funcionalidad	
Fecha	
Fuente	
Tipo	
Necesidad	
Estado	
Descripción	
Resultado emulador Android Studio	
Resultado LG G3S	

Tabla 5. Ejemplo tabla de casos de prueba

En ella se observan 10 campos cuya descripción se detalla a continuación con el fin de su correcta comprensión en su lectura:

- **ID:** cada caso de prueba será identificado unívocamente mediante este campo. Al ser un número reducido de requisitos no se dividen bajo ninguna otra característica de los mismos y se llamarán CPX donde CP hace referencia a que es un Caso de Prueba y en lugar de la X aparece un número, estos números se otorgarán de forma secuencial al lugar que ocupan en esta memoria.
- **Funcionalidad:** representa brevemente cuál es la utilidad que se quiere probar sobre la aplicación.
- **Fecha:** tiempo en que se realizó la prueba.
- **Fuente:** indica quién ha descrito el caso de prueba al considerar que era necesario probarlo.
- **Tipo:** indica si se trata de un caso de prueba funcional, qué debe hacer la aplicación, o no funcional, cómo deben ejecutarse las funcionalidades de la aplicación.
- **Necesidad:** indica si el resultado positivo o negativo del caso de prueba es una funcionalidad básica para que el objetivo con el que se desarrolla la aplicación se alcance. Los valores que puede tomar son:
 - Esencial: mostrando que si debe superar la prueba para conseguir el objetivo global.
 - Deseable: mostrando que puede no superar la prueba y el objetivo inicial de la aplicación se conseguirá lograr.
- **Estado:** indica si el caso de prueba se ha realizado o es un caso de prueba que se realizará en un futuro. Los valores que puede tomar este campo son Aplicado/No Aplicado.
- **Pasos del caso de prueba:** en este campo se detallan los pasos que se deben seguir para comprobar si la prueba se realiza con éxito.
- **Resultado emulador Android Studio:** indica el resultado de la prueba en el emulador de la herramienta utilizada para el desarrollo de la aplicación.
- **Resultado LG G3S:** indica el resultado de la prueba en el dispositivo móvil seleccionado.

A continuación se detallan los 11 casos de prueba seleccionados para probar las funcionalidades de la aplicación y otros aspectos no funcionales:

ID	CP-1
Funcionalidad	Registro en la aplicación
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	<p>Abrir la aplicación.</p> <p>Pulsar sobre el botón REGISTRARSE para acceder a la pantalla de registro.</p> <p>Rellenar todos los campos requeridos.</p> <p>Pulsar sobre REGISTRO.</p> <p>Visualizar pantalla que muestra que el registro se ha realizado con éxito.</p>
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 6. Caso de prueba 1: Registro en la aplicación

ID	CP-2
Funcionalidad	Inicio de sesión en la aplicación
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	<p>Abrir la aplicación.</p> <p>Rellenar los campos email y contraseña.</p> <p>Pulsar sobre INICIAR SESIÓN.</p> <p>Comprobar que se accede a la pantalla de las funcionalidades de la aplicación.</p>
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 7. Caso de prueba 2: Inicio de sesión en la aplicación.

ID	CP-3
Funcionalidad	Inserción de pregunta
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	<p>Iniciar sesión en la aplicación.</p> <p>Pulsar sobre INSERTAR PREGUNTA.</p> <p>Rellenar todos los campos requeridos mediante el teclado o la voz.</p> <p>Pulsar sobre GUARDAR PREGUNTA.</p> <p>Visualizar pantalla que muestra que la pregunta se ha guardado correctamente.</p>
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 8. Caso de prueba 3: Inserción de pregunta.

ID	CP-4
Funcionalidad	Realización de test
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	<p>Iniciar sesión en la aplicación.</p> <p>Pulsar sobre TEST.</p> <p>Pulsar sobre PREGUNTAS TEST.</p> <p>Pulsar la respuesta que se cree que es la correcta.</p> <p>Pulsar sobre SIGUIENTE.</p> <p>Contestar las preguntas que van apareciendo secuencialmente en la pantalla.</p> <p>Pulsar sobre FINALIZAR.</p> <p>Visualizar una pantalla con el número de aciertos.</p>
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 9. Caso de prueba 4: Realización de TEST.

ID	CP-5
Funcionalidad	Realización de pregunta abierta
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Deseable
Estado	Aplicado
Pasos	<p>Iniciar sesión en la aplicación.</p> <p>Pulsar sobre TEST.</p> <p>Pulsar sobre PREGUNTA ABIERTA.</p> <p>Escribir o decir mediante la voz la respuesta.</p> <p>Pulsar sobre FINALIZAR.</p> <p>Visualizar una pantalla con el porcentaje de acierto.</p>
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 10. Caso de prueba 5: Realización de pregunta abierta.

ID	CP-6
Funcionalidad	Edición de pregunta guardada
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Deseable
Estado	Aplicado
Pasos	<p>Iniciar sesión en la aplicación.</p> <p>Pulsar sobre VER PREGUNTAS.</p> <p>Pulsar sobre una pregunta de la lista.</p> <p>Visualizar los campos de pregunta, respuestas y palabras clave pudiendo ser editados.</p> <p>Pulsar sobre EDITAR PREGUNTA.</p> <p>Visualizar una pantalla en la que se muestra que la pregunta ha sido modificada correctamente.</p>
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 11. Caso de prueba 6: Edición de pregunta guardada.

ID	CP-7
Funcionalidad	Visualización de progreso y estadísticas
Fecha	15/09/2016
Fuente	Analista
Tipo	Funcional
Necesidad	Deseable
Estado	Aplicado
Pasos	Iniciar sesión en la aplicación. Pulsar sobre VER PROGRESIÓN. Visualizar dos tablas con la misma estructura: número de preguntas realizadas, número de preguntas acertadas, número de preguntas falladas y el porcentaje de acierto. La primera tabla muestra las preguntas tipo test y la segunda las preguntas abiertas.
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 12. Caso de prueba 7: Visualización de progreso y estadísticas.

ID	CP-8
Funcionalidad	La aplicación detecta la pantalla táctil
Fecha	15/09/2016
Fuente	Analista
Tipo	No Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	Acceder a la aplicación. Navegar por la aplicación mediante el táctil de la pantalla. Comprobar que reconoce a la perfección el touch en la pantalla.
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 13. Caso de prueba 8: La aplicación detecta la pantalla táctil.

ID	CP-9
Funcionalidad	La aplicación detecta el teclado
Fecha	15/09/2016
Fuente	Analista
Tipo	No Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	Acceder a la aplicación. Pulsar sobre INSERTAR PREGUNTA. Pulsar sobre alguno de los campos que permite la escritura. Teclear un texto mediante el teclado. Visualizar que las teclas se introducen en la pantalla.
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 14. Caso de prueba 9: La aplicación detecta el teclado.

ID	CP-10
Funcionalidad	La aplicación detecta el reconocimiento de voz
Fecha	15/09/2016
Fuente	Analista
Tipo	No Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	Acceder a la aplicación. Pulsar sobre INSERTAR PREGUNTA. Pulsar sobre alguno de los campos que permite la escritura. Hablar al dispositivo. Visualizar que la voz se transcribe en la pantalla.
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

Tabla 15. Caso de prueba 10: La aplicación detecta el reconocimiento de voz.

ID	CP-11
Funcionalidad	Interfaz intuitiva, amigable y usable
Fecha	15/09/2016
Fuente	Analista
Tipo	No Funcional
Necesidad	Esencial
Estado	Aplicado
Pasos	Acceder a la aplicación. Navegar durante 5 minutos por sus pantallas. Comprobar que se ha podido acceder al recurso deseado en cada momento sin problemas. Comprobar que no la interfaz no ha producido rechazo en el usuario.
Resultado emulador Android Studio	OK
Resultado LG G3S	OK

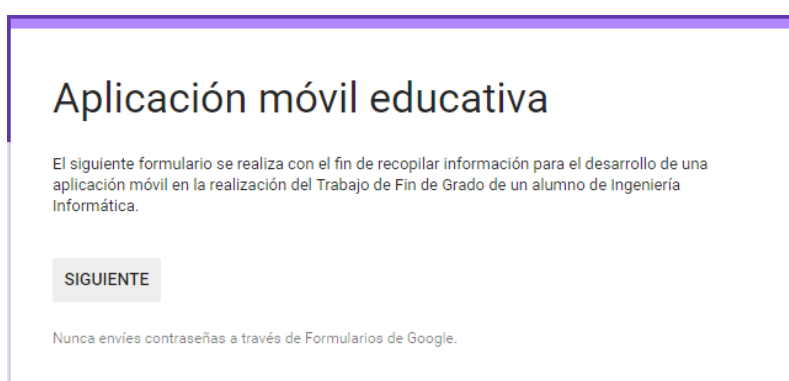
Tabla 16. Caso de prueba 11: Interfaz intuitiva, amigable y usable.

4.3 Encuesta a usuarios potenciales

Con el fin de comprobar si la aplicación desarrollada cumpliría lo demandado por los usuarios, se ha realizado una encuesta online a través de los [formularios de Google](#) de esta forma se podría recopilar de manera rápida y barata toda la información posible de una muestra variada de clientes potenciales. Así se sabría que se ha realizado bien y en que se debe incidir en futuras actualizaciones de la aplicación.

4.3.1 Preguntas realizadas

La encuesta se divide en secciones para poder guiar a los encuestados de una forma u otra en función de sus respuestas, es decir, dependiendo de sus respuestas realizarán la encuesta completa o solamente las secciones en las que sus datos sean significativos (Por ejemplo, una persona que no utiliza su dispositivo móvil para descargar aplicaciones no tiene sentido que continúe respondiendo al resto de preguntas pues sus respuestas no tendrán el mismo valor que la de encuestados que si lo utilizan para ese determinado fin)



Aplicación móvil educativa

El siguiente formulario se realiza con el fin de recopilar información para el desarrollo de una aplicación móvil en la realización del Trabajo de Fin de Grado de un alumno de Ingeniería Informática.

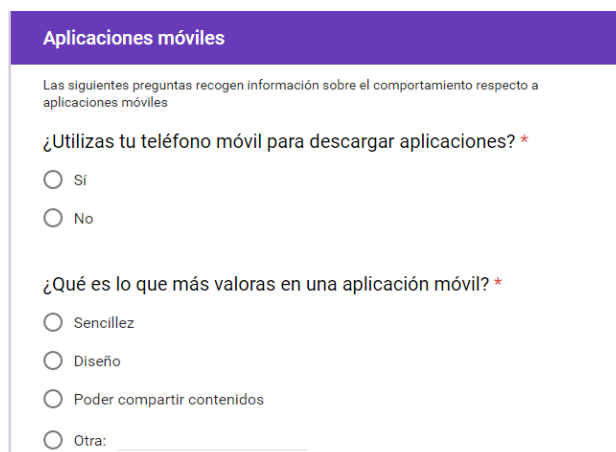
SIGUIENTE

Nunca envíes contraseñas a través de Formularios de Google.

Figura 24. Preguntas encuesta: Introducción a usuarios

- **SECCIÓN 1: Comportamiento frente a aplicaciones móviles**

En esta sección se quiere comprobar que el encuestado utiliza su teléfono móvil para descargar aplicaciones. En caso positivo se trata de un encuestado objetivo y continúa la encuesta de forma normal, de lo contrario se le manda a las preguntas de datos estadísticos para incluir su perfil como usuario que no utilizaría la aplicación. Además, se pregunta cuál es la característica que más valora en una aplicación con el fin de conocer si el desarrollo ha sido exitoso o si habría que mejorar la aplicación e incidir en otro aspecto que los usuarios valoren más.



Aplicaciones móviles

Las siguientes preguntas recogen información sobre el comportamiento respecto a aplicaciones móviles

¿Utilizas tu teléfono móvil para descargar aplicaciones? *

☐ Sí

☐ No

¿Qué es lo que más valoras en una aplicación móvil? *

☐ Sencillez

☐ Diseño

☐ Poder compartir contenidos

☐ Otra: _____

Figura 25. Preguntas encuesta: Comportamiento de los usuarios sobre aplicaciones móviles

- **SECCIÓN 2: Hábitos de estudio del usuario**

Una vez realizada la primera criba de encuestados y antes de preguntar al usuario por temas concretos sobre la aplicación, se debe contestar a 2 preguntas sobre hábitos de estudios: qué se utiliza para estudiar y desde donde se repasan los conocimientos adquiridos. Estas preguntas son útiles para conocer si sería una aplicación que los estudiantes demandaran. Finalmente se pregunta directamente al usuario si utilizaría una aplicación de apoyo en el repaso, si la respuesta es afirmativa continúa secuencialmente la encuesta, por el contrario si su respuesta es negativa es conducido a las preguntas de datos estadísticos.

Hábitos del usuario

Las siguientes preguntas recogen información del comportamiento del usuario durante su estudio.

¿Que recursos utilizas para estudiar? *

☐ Libros

☐ Apuntes

☐ Recursos online

☐ Aplicaciones móviles

☐ Otro: _____

¿Cómo sueles repasar los contenidos estudiados? *

☐ Revisando mis apuntes

☐ Le pido a alguien que me pregunte el temario

☐ Realizo un esquema en papel

☐ Lo digo en voz alta

☐ Otra: _____

¿Utilizarías una aplicación móvil que te ayudara a repasar los contenidos estudiados? *

☐ Sí

☐ No

Figura 26. Preguntas encuesta: Hábitos del usuario durante el estudio

- **SECCIÓN 3: Qué contenidos les gustaría que tuviera una aplicación móvil de preguntas-respuesta:**

Se pregunta al usuario su opinión acerca de las funcionalidades con las que partirá la versión 1.0 de la aplicación para comprobar si lo desarrollado por el analista programador es lo que buscan los usuarios potenciales en una aplicación de estas características, o si por el contrario no se alinea con las necesidades de los usuarios y por tanto se deben desarrollar otras funcionalidades. Como ya se ha comentado, gracias a las preguntas de clasificación, a esta sección solo llegarán usuarios que utilizan sus teléfonos móviles para descargar aplicaciones y que, además, querrían una aplicación que les ayudara a repasar los contenidos estudiados. Estas preguntas van desde características funcionales hasta estéticas:

Contenidos aplicación educativa

La aplicación que se va a desarrollar busca que el usuario pueda repasar contenidos que previamente ha estudiado. En las siguientes preguntas se recoge información sobre cómo le gustaría al usuario que funcionara la aplicación.

¿Cómo te gustaría acceder a los contenidos? *

☐ Introducirlos yo mismo
☐ Utilizar los de otro usuario
☐ Ambas
☐ Otra: _____

¿Cómo querrías que te evaluara esa aplicación? *


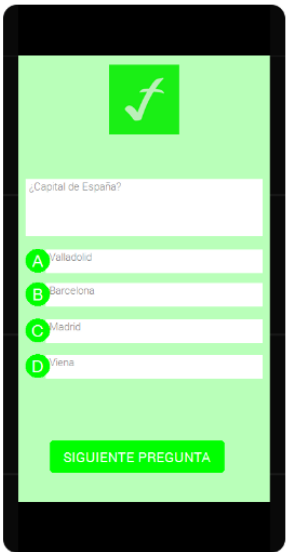
☐ Preguntas tipo TEST
☐ Preguntas de desarrollo
☐ Ambas
☐ Otra: _____

¿Te gustaría poder ver tu progreso y estadísticas de tus preguntas? *

☐ Sí
☐ No

Figura 27. Preguntas encuesta: Contenidos aplicación educativa (1 de 2)

¿Te gusta el siguiente diseño para una aplicación educativa de repaso de contenidos? *

☐ Sí
☐ No
☐ Otra: _____

¿Qué más te gustaría que tuviera la aplicación?

Tu respuesta

Figura 28. Preguntas encuesta: Contenidos aplicación educativa (2 de 2)

- **SECCIÓN 4: Datos estadísticos para poder analizar la muestra mediante la edad y los estudios de los encuestados:**

En esta sección se busca clasificar a los encuestados en relación a su edad, estudios y el sistema operativo de su teléfono móvil. Inicialmente, esta aplicación está pensada para estudiantes de educación secundaria e instituto, pero se quiere conocer si otro tipo de estudiantes estaría interesado en ella por si sería necesario incluir algún que otro aspecto que podría ser útil. Además, una pregunta clave en esta encuesta es el sistema operativo que utiliza el usuario en su teléfono móvil ya que al realizar un estudio teórico sobre los sistemas operativos, se conoció que alrededor del 75% de los usuarios pertenecían a Android, menos del 25% a iOS y lo restante a BlackBerry y Windows Phone. Esta pregunta es muy importante para conocer si el desarrollo en Android está correctamente enfocado pues se eligió como SO para la primera versión por poder llegar a un público más amplio.

Datos estadísticos

La siguiente información se recoge para clasificar a los entrevistados en distintos grupos acordes a su edad y nivel de estudios.

Edad *

☐ Menor de 18 años

☐ Entre 18 y 30 años

☐ Mayor de 30 años

Máximo nivel de estudios alcanzado o cursando *

☐ ESO

☐ Bachillerato

☐ Formación Profesional

☐ Universitarios

☐ Otra: _____

Sistema operativo de tu teléfono móvil *

☐ Android

☐ iOS

☐ Blackberry

☐ Windows Phone

☐ Otra: _____

Figura 29. Preguntas encuesta: Datos estadísticos de los encuestados

4.3.2 Respuestas obtenidas y conclusiones extraídas

Después de permitir a los usuarios contestar libremente a la encuesta durante un período limitado de tiempo, se consiguió la respuesta de 72 personas. En base a sus respuestas, se han podido comprobar algunas de las conclusiones obtenidas tras estudiar en el estado del arte aplicaciones similares y además, se han extraído respuestas generales sobre las que se deben implementar mejoras en la aplicación con el fin de llegar a un mayor número de usuarios y cumplir sus expectativas.

Se comienza comentando los resultados obtenidos en la sección 4 de la encuesta puesto que los datos estadísticos se incluyen al finalizar la encuesta debido a que si se preguntara al principio por ello se podría obtener un rechazo del usuario, sin embargo su evaluación es necesaria al principio para clasificar a los encuestados en relación a estos datos. La muestra de 72 personas es variada en cuanto a la edad de los participantes, sus estudios y el sistema operativo de su teléfono móvil, aunque predomina la gente joven de entre 18 y 30 años, con estudios universitarios y quienes disponen de Android como sistema operativo en su teléfono móvil. El perfil por tanto, a priori, sería el público objetivo al que va dirigida la aplicación: gente joven con estudios y con Android en sus dispositivos. Por tanto, podrán descargar la aplicación una vez se lance al mercado. De las preguntas que recogen datos estadísticos, sería interesante mostrar un gráfico circular que divide a la muestra por el SO de sus móviles:

Sistema operativo de tu teléfono móvil (72 respuestas)

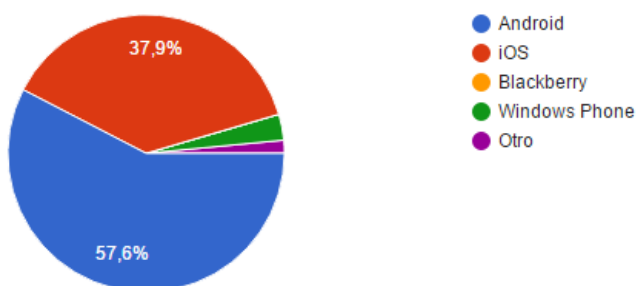


Figura 30. Respuestas encuesta: SO más utilizado

Contrastando estos resultados con los extraídos de internet al iniciar el TFG para decidir la plataforma en la que se desarrollaría la aplicación, se comprueba que el sistema operativo más usado es Android, pero con menor diferencia de iOS que la facilitada online: 57,6% prefieren el primero en lugar de alrededor del 75%, y un 37,9% se decantan por tener un dispositivo marca Apple en lugar del casi 25% encontrado. Aunque los datos no coincidan exactamente, el orden de preferencia es el mismo y con una distancia considerable entre ambos.

Se continúa observando los resultados de la sección 1 la cual muestra que un 95,8% utiliza su teléfono móvil para descargar aplicaciones. Solamente 3 personas de las 72 que componen la muestra no lo hacen y tienen las siguientes características:

¿Utilizas tu teléfono móvil para descargar aplicaciones?	¿Qué es lo que más valoras en una aplicación móvil?	Edad	Máximo nivel de estudios alcanzado	Sistema operativo de tu teléfono móvil
No	Que ocupe poca memoria	Mayor de 30 años	Universitarios	Android
No	Sencillez	Mayor de 30 años	Universitarios	Android
No	Sencillez	Entre 18 y 30 años	Universitarios	Android

Figura 31. Respuestas encuesta: Usuarios que no usan apps móviles

Entre los 69 restantes que contestaron afirmativamente, un 68,1% valoran sobretodo la sencillez de la aplicación sobre otras características:

- 15,3% valoran otra cualidad. Esta opción permitía al usuario contestar abiertamente, entre esas respuestas destacan que la aplicación sea: útil, rápida, ligera en espacio y en uso de datos. La utilidad se asemeja con la funcionalidad, y las otras opciones que señalan los usuarios son aspectos no funcionales pero que se deben tener en cuenta en la medida de lo posible. Mientras que la rapidez con la que funcione es un aspecto que depende del móvil del usuario, el control del espacio y de datos si sería una característica a tener en cuenta en el desarrollo utilizando bases de datos en la nube para eliminar el problema del espacio pero que aumentaría el consumo de datos, si se mejora una cualidad se empeora otra, habría que encontrar un equilibrio como utilizar una base de datos que cargue los contenidos a la nube cuando esté conectada a wifi y solo recupere los necesarios al utilizar la aplicación.
- 11,1% valora el diseño de la aplicación, se debe satisfacer al usuario otorgando un diseño amigable con el usuario como fue una de las principales características al desarrollar el prototipo.
- En último lugar, con un 5,6% y solo 4 usuarios como preferencia, eligen la compartición de contenidos, es decir, componente de red social en la aplicación. En la implementación se ha relegado a un segundo plano ese componente para centrar la funcionalidad meramente en la inserción de preguntas y el poner a prueba conocimientos. Para futuras versiones de la aplicación se tendrá en cuenta con el fin de satisfacer al mayor número de usuarios y ofrecer una característica tan importante como esa.

La sección 2 pregunta al usuario sobre sus hábitos de estudio para saber si el desarrollo de la aplicación podrá ser un elemento más de apoyo a su estudio.

La primera pregunta permite seleccionar varios recursos de una lista a cada encuestado:

¿Que recursos utilizas para estudiar? (69 respuestas)

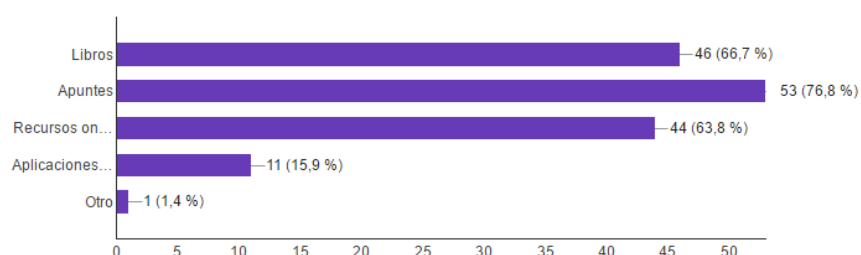


Figura 32. Respuestas encuesta: Recursos utilizados por los usuarios

El estudio mediante apuntes encabeza la primera opción de la lista, mientras que el segundo lugar se reparte entre libros y recursos online. Actualmente solo 11 de los 69 encuestados, es decir un 15,9%, utiliza aplicaciones móviles para estudiar. Esta pregunta no tiene un valor significativo para el desarrollo de la aplicación puesto que el enfoque es repaso del estudio. Sin embargo, muestra que aún se siguen utilizando métodos tradicionales como libros y apuntes.

La pregunta sobre como suele repasar los contenidos sí que tiene un gran valor para el estudio de los clientes potenciales pues arroja datos sobre lo que quieren los usuarios.

- 44,9% → En primer lugar los encuestados hacen un diagrama en papel, esta opción podría asemejarse a la funcionalidad de pregunta abierta que ofrece la aplicación, puesto que puede escribir sin límites lo que desee y recibirá un porcentaje de acierto.
- 34,8% → Como segunda opción, los usuarios revisan sus apuntes, sería interesante entonces incluir una pestaña sobre la que los usuarios puedan adjuntar documentos en PDF, WORD y TXT para que depositen sus apuntes en caso de quererlo, y así poder consultarlos en cualquier momento.
- 20,3% → En tercer y último lugar, se repiten en voz alta lo estudiado, igual que la preferencia mayoritaria, la aplicación permite hablar al teléfono móvil, que lo reconocerá y transcribirá, y devolverá un porcentaje de acierto.

La última pregunta de esta sección obtiene un 85,5% de usuarios que sí utilizarían una aplicación móvil para repasar los contenidos estudiados. 59 de los 69 encuestados pasan la segunda criba y son los usuarios potenciales que se someterán a preguntas concretas sobre funcionalidades.

Por último, se comenta la sección 3 que es la más importante y significativa de la encuesta puesto que recoge información sobre si las principales características implementadas son valoradas por los clientes potenciales.

A los usuarios les gustaría poder introducir sus propios contenidos y usar los de otro usuario. En la primera versión solo podrán disfrutar de sus propios contenidos, pero como demostró otra pregunta de la encuesta, el componente social requerido por los usuarios, el cual sería necesario para la opción de poder utilizar contenidos que otros compartan. Es un punto que se tiene que tener en cuenta en próximos desarrollos.

Preguntados sobre las opciones de ser evaluados, los usuarios querrían en mayor medida contestar preguntas de desarrollo y tipo TEST. Como segunda opción, solamente de tipo TEST. El desarrollo cubre ambas opciones, se extrae de esta pregunta que la opción más usada será evaluarse con preguntas cerradas.

Todos los usuarios excepto 1 quieren ver su progreso y estadísticas, la inmensa mayoría reclama esta posibilidad, que se cubre con la versión 1.0 por lo que los usuarios terminarán su repaso satisfechos con una tabla que lo muestre.

La penúltima pregunta muestra a los usuarios 2 pantallas de la aplicación para comprobar si les gusta el diseño para una app de repaso de contenidos. La respuesta es positiva en un 86,4% mientras que un 13,6% rechazan la vista. 2 personas además, opinan lo siguiente sobre cómo mejorar el diseño: utilizar unos colores menos llamativos puesto resaltan demasiado en la pantalla.

Finalmente, se concluye la encuesta con una pregunta abierta y opcional en la que se pide conocer que funcionalidades desearían que tuviera. Esta pregunta es muy importante para poder desarrollar una estrategia en el desarrollo del trabajo futuro pues los usuarios han opinado sobre qué es lo que quieren. Se han conseguido 20 opiniones:

Ayuda y acceso al temario en caso de no saber contestar una pregunta
Que te diera más información en caso de que fallaras (no solo cuál era la correcta)
Sistema de puntuación
Recopilatorio de respuestas falladas para poder responder las bien después
Preguntas de desarrollo y poder hacer esquemas
El color es demasiado fuerte para estudiar por lo que agotaría de manera más rápida la atención del usuario. Recomendando el uso de colores más suaves y agradables, que no cansen tanto la vista.
Incluir a la pregunta el grado de dificultad de la pregunta, para después poder elegir con qué dificultad se desea repasar.
Imágenes
Algún tipo de opción para interactuar con otros compañeros
Reseñas, chat con otros usuarios, sección de ayuda
Contenido extra
Poder retar a compañeros o compartir tus resultados. Incluso poder lanzar una pregunta a compañeros y ver los resultados globales de las respuestas de todos los estudiantes que han contestado.
Enlazar contenidos
Poder compartir temas
Recomendaciones para mejorar
Preguntas introducidas por el profesor o alguna fuente más fiable que otros alumnos (en dso se veían preguntas muy raras), también estaría bien que incluyera apuntes sin más, estaría bien tener un foro de preguntas, y que la app en general estuviera hecha a nivel de asignatura con el profesor involucrado más que alumnos solos
Un concurso con otros alumnos para ver quién se lo sabe mejor
Audios con las lecciones
poder decir la lección como si fuera un profesor recién grabado
No sólo preguntas con texto, también preguntas con esquemas/dibujos

Figura 33. Respuestas encuesta: Recomendaciones de mejora de los usuarios

En el siguiente párrafo se resumen estas respuestas traducidas a nuevas funcionalidades:

- Pestaña en la que guardar apuntes.
- Enlazar las preguntas a partes de los apuntes para ver por qué se ha fallado.
- Histórico de respuestas falladas.
- Poder tomar notas y realizar esquemas.
- Como característica no funcional, se deben utilizar colores menos llamativos que no cansen a la vista
- Incluir dificultad de preguntas para poder seleccionar la complejidad de los test.
- Inclusión de imágenes.
- Componente social: chat con otros usuarios, reseñas, sección de ayuda, retos a compañeros, compartición de resultados, ver ranking de usuarios.
- Roles que diferencien a alumnos de profesores.

Las conclusiones obtenidas de esta encuesta, además de la inclusión de la lista de funcionalidades mencionadas en el párrafo anterior son las siguientes:

- El sistema operativo más utilizado es Android.
- El principal aspecto valorado por los usuarios es la sencillez. Una aplicación intuitiva y fácil de usar recibirá un mayor respaldo.
- Los usuarios quieren poder escribir/decir sus respuestas abiertamente sin tener que contestar a una respuesta prefabricada.
- Los usuarios si utilizarían una aplicación educativa de estas características.
- Los usuarios quieren crear sus contenidos y utilizar los de otros. Se debe desarrollar el componente social.
- Los usuarios quieren preguntas tipo test y preguntas abiertas pero prefieren las de tipo test.
- Ver el progreso es una característica fundamental.
- Se debe rebajar la intensidad de los colores de la aplicación.

CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO

Finalmente, tras haber desarrollado la aplicación y haberla puesto a prueba además de haber hecho una encuesta a usuarios potenciales, en este punto se recogen los resultados obtenidos y se hace una retrospectiva de ellos para comprobar si se han conseguido los objetivos deseados. Se exponen las conclusiones finales tras haber realizado el trabajo y se enuncia el trabajo futuro de la misma para poder extender las funcionalidades y que la aplicación llegue a un público más amplio y cumpla sus requisitos.

5.1 Conclusiones

Este TFG recoge el estudio para el desarrollo de una aplicación para dispositivos móviles cuyo SO fuera Android. Pudiendo interaccionar con ella vía táctil o mediante la voz. La aplicación en cuestión permite a los usuarios poner a prueba los conocimientos adquiridos gracias a preguntas tipo test y preguntas de desarrollar cuyas respuestas cargaron ellos previamente. El usuario puede navegar entre las distintas pantallas de forma táctil y responder a las preguntas de forma táctil u oral.

La primera tarea consistió en definir el alcance de la aplicación y de la memoria objeto de TFG. Se tardó más tiempo del debido en comenzar seriamente con el proyecto por lo que hubo fechas que se retrasaron de la planificación inicial. Además la fase de desarrollo ocupó más tiempo del estimado por lo que se ha aprendido a organizar el tiempo de forma más ordenada y planificar la fecha de entrega con más antelación, fijando incluso un tiempo para imprevistos.

Inicialmente, se estudiaron aplicaciones similares que estuvieran en el mercado móvil llegando a la conclusión que los usuarios requerían una aplicación simple y que les permitiera ponerse a prueba de sus conocimientos. Una interfaz sencilla e intuitiva, en la que los colores no cansaran al usuario, y sumado a funcionalidades básicas harían que la aplicación fuera demandada por los usuarios.

La fase de desarrollo ha sido la más complicada ya que se encontraron numerosas trabas a la hora de descargar la herramienta principal de implementación: Android Studio. Debido al SDK necesario para poder poner en funcionamiento la aplicación y rutas de acceso cuyo nombre contenía una tilde se tuvo que buscar información para la resolución de errores a través de internet. Sin embargo, gracias a que es una herramienta muy usada, el nivel de información que se encuentra en la red en páginas como foros de colaboración era alta por lo que se pudieron subsanar los errores.

Una vez se tuvo preparado el escritorio de trabajo, se comenzaron desarrollando las interfaces siguiendo el prototipo diseñado en la fase de planificación y tutoriales en forma de vídeo y escrito. Fue posible completar esta subfase en un corto periodo de tiempo. Cuando se pasó a implementar la lógica de negocio, lo primero fue vincular los controles de las interfaces, capa de presentación (vista), con la que sería la capa de lógica (controlador) para ello se siguieron indicaciones en foros y a pesar de ser una tarea laboriosa por el elevado número de controles en cada pantalla, se convirtió en un proceso mecánico.

Acabada esta tarea, llegó la subfase más complicada ante la que se enfrentó el desarrollador: la lógica de negocio. Se debía implementar la capa núcleo que comunicaría los datos contenidos en la base de datos con las pantallas de la aplicación. Se codificó un conector para poder enviar órdenes a la capa de persistencia (modelo) y se desarrolló el código de cálculo con bastantes problemas y gran ocupación del tiempo disponible antes de la entrega. Paralelamente, se montó una base de datos en la herramienta MySQL dónde se guardarían los datos, esta tarea fue sencilla ya que a lo largo del grado realizado se hizo varias veces en distintas asignaturas.

Tras montar la base de datos, se construyeron las consultas que se lanzarían contra ella y cuyos campos variables (preguntas, respuestas, palabras clave, etc.) vendrían definidos desde la capa de lógica de negocio que las recogería desde la capa de presentación dónde las definiría el usuario.

Finalmente, en la fase de documentación hubo que apurar hasta el último minuto para poder comentar todo lo ocurrido durante el desarrollo del TFG y dejarlo en un formato claro y formal.

En esta fase, se realizaron pruebas a la aplicación cuyo resultado fue satisfactorio y una encuesta a clientes potenciales que reveló que la mayoría de las ideas sobre las que se enfocó el desarrollo eran bien recibidas, pero requerían más funcionalidades, las que se incluirán como parte del trabajo futuro.

5.2 Trabajo Futuro

A continuación se exponen las líneas futuras de investigación sobre la aplicación desarrollada:

- La principal necesidad es expandir al segundo sistema operativo preferido por los usuarios la aplicación. Para ello se debe investigar el mercado de Apple y si sería un proyecto rentable.
- En segundo lugar, sería conveniente destacar las funcionalidades futuras que va a tener la aplicación. Estas funcionalidades están desarrolladas y resumidas por la encuesta realizada a los usuarios:
 - Pestaña en la que guardar apuntes.
 - Enlazar las preguntas a partes de los apuntes para ver por qué se ha fallado.
 - Histórico de respuestas falladas.
 - Poder tomar notas y realizar esquemas.
 - Como característica no funcional, se deben utilizar colores menos llamativos que no cansen a la vista
 - Incluir dificultad de preguntas para poder seleccionar la complejidad de los test.
 - Inclusión de imágenes.
 - Componente social: chat con otros usuarios, reseñas, sección de ayuda, retos a compañeros, compartición de resultados, ver ranking de usuarios.
 - Roles que diferencien a alumnos de profesores.
- Al ser una aplicación con poco texto predefinido, sería sencillo traducirla a otros idiomas para ampliar el público objetivo.
- Con distintos colores predefinidos en las hojas de estilo, se definirían más paletas de colores, además de la verde actual, para que los distintos usuarios puedan seleccionar su color favorito entre una gama cerrada.
- Mejorar la funcionalidad de pregunta abierta ya que solo son posibles 6 notas y funciona si la afirmación es negativa como positiva ya que se fija en literales únicos y no combinación de palabras.

GLOSARIO

- **3G:** El 3G, o tercera generación, es una tecnología móvil que permite la transmisión de datos, voz y vídeo a una alta velocidad y sin cables.
- **3GP:** formato de video.

A

- **ADE:** siglas para Administración y Dirección de Empresas.
- **API:** interfaz de programación de aplicaciones.
- **Android:** sistema operativo que funciona en dispositivos móviles.
- **Android Studio:** herramienta con la que diseñar aplicaciones para dispositivos con sistema operativo android.
- **App:** aplicación o programa informático que se utiliza en dispositivos móviles.

B

- **BBDD:** Siglas para Base de Datos
- **Blackberry:** sistema operativo para dispositivos móviles.
- **Builder:** complemento para el diseño de interfaces.
- **B-learning:** siglas para Blended Learning, que se refiere a aprendizaje semipresencial.

C

- **Chatear:** comunicarse a través de un chat, que es un servicio de mensajería instantánea online
- **Comando Delete:** comando utilizado para borrar un registro en una base de datos.
- **Comando Update:** comando utilizado para actualizar un registro en una base de datos.
- **Compilador:** programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje diferente para adaptarlo a su uso.
- **Copy&Paste:** concepto proveniente del inglés utilizado para referirse a realizar acciones de copia y pega.

D

- **DBMS:** siglas para database Management System, que en español significa Sistema Gestor de Bases de Datos (SGBD).
- **Driver:** Controlador, rutina o programa que enlaza un dispositivo periférico al sistema operativo.

E

- **E/S:** Siglas para Entrada/Salida.
- **EMMA:** Siglas para Extensible multimodal Annotation, que en español quiere decir: Extensibilidad multimodal anotaciones.
- **Enterprise:** tecnicismo utilizado para referirse a empresa.
- **E-learning:** siglas para Electronic Learning, que se refiere a aprendizaje electrónico.
- **E-mail:** correo electrónico.

F

- **Facebook:** red social.

G

- **GANTT**: herramienta utilizada para crear diagramas de fijación de tiempos.
- **GNU**: sistema operativo libre.
- **GPS**: sistema de posicionamiento global.
- **Google**: compañía especializada en productos y servicios relacionados con internet, software y tecnología en general.
- **Google +**: red social de la compañía google.

H

- **H2, H6 y H8**: siglas para referirse a la segunda, sexta y octava heurística de nielsen.
- **HTTPS**: siglas para Hypertext Transfer Protocol Secure, en español Protocolo de transferencia de hipertexto.
- **Heurísticas de Nielsen**: recomendaciones para crear sistemas amigables para el usuario.
- **Holo**: interfaz propia de android.
- **Hotspot**: tecnicismo proveniente del inglés para referirse a un lugar que ofrece acceso a internet a través de una red inalámbrica.

I

- **IDE**: siglas para referirse a Integrated Development Environment, que en español significa Entorno de Desarrollo Integrado.
- **INKML**: lenguaje XML.
- **iPod**: dispositivo reproductor de música.
- **iOS**: Sistema operativo para móviles Apple.

J

- **JIT**: siglas para Just in time que se refieren al tiempo de ejecución.
- **Java**: lenguaje de programación.
- **Justinmind**: herramienta utilizada para el prototipado de interfaces.

L

- **Layout**: capa sobre la que se diseña una interfaz.
- **Link**: enlace a un sitio web.

M

- **MicroSD**: tarjeta extraíble.
- **M learning**: proyecto creado para ayudar a personas que no desean recibir educación formal.
- **MPEG-4**: formato de vídeo.
- **MVC**: modelo arquitectónico software para Modelo-Vista-Controlador.
- **Mac OS**: sistema operativo de dispositivos Apple.
- **Market**: mercado online para comprar apps.
- **Microsoft Windows**: sistema operativo.
- **Mobile learning**: concepto que se refiere al aprendizaje gracias a las tecnologías móviles.
- **Multitouch**: sistema que permite que un dispositivo reconozca varios movimientos a la vez.
- **Mysql**: herramienta utilizada para la creación de una base de datos.

N

- **NFC:** siglas para Near Field Communication, en español Comunicación de campo cercano.

P

- **PDA:** siglas para Personal Digital Assistant, en español Asistente Digital Personal.
- **Persistencia:** referido a la firmeza de información en la base de datos.
- **Plugin:** complemento que añade valor a una herramienta.
- **Pocket PC:** ordenador de bolsillo.

S

- **SDK:** siglas para Software Development Kit, en español Kit de Desarrollo Software.
- **SGBD:** siglas para Sistema Gestor de Base de Datos.
- **SISR:** siglas para Interpretación semántica del reconocimiento del habla.
- **SO:** siglas para Sistema Operativo.
- **SQL:** lenguaje de programación.
- **SRGS:** siglas para Gramática de reconocimiento del habla.
- **SSML:** siglas para Lenguaje de Marcado de Síntesis del Habla.
- **Skype:** plataforma de comunicación online.
- **Smart devices:** tecnicismo para referirse a dispositivos inteligentes.
- **Smartphone:** tecnicismo para referirse a teléfonos inteligentes.

T

- **TFG:** siglas para Trabajo de Fin de Grado
- **Tething:** proceso por el cual un dispositivo con conexión a internet ofrece conexión a otros dispositivos.

U

- **Ubuntu Linux:** Sistema operativo
- **URL:** enlace a sitio web.

V

- **Voicexml:** lenguaje de etiquetado.

W

- **W3C:** siglas para World Wide Web Consortium, que es una entidad que genera recomendaciones y estándares.
- **Web:** sitio que está en internet.
- **Wifi:** tecnología para ofrecer internet bajo redes inalámbricas.
- **Windows Phone:** Sistema operativo para teléfonos Windows.

X

- **Xhdpi:** resolución que soporta múltiples pantallas.
- **XML:** lenguaje de programación.

Y

- **Youtube:** plataforma de vídeos.

Bibliografía

- [1] Xataka. (Junio de 2016). *xataka*. Obtenido de xataka:
<http://www.xataka.com/moviles/gartner-el-98-del-mercado-movil-es-de-ios-y-android-crecio-un-15-gracias-a-los-mercados-emergentes>
- [2] Gobierno de España. (Junio de 2016). *recursostic*. Obtenido de recursostic:
<http://recursostic.educacion.es/observatorio/web/fr/cajon-de-sastre/38-cajon-de-sastre/1026-movil-learning>
- [3] AulaPlaneta. (Junio de 2016). *AulaPlaneta*. Obtenido de
<http://www.aulaplaneta.com/2015/04/07/recursos-tic/las-mejores-herramientas-para-crear-cuestionarios-interactivos/>
- [4] UGR. (Julio de 2016). *UGR*. Obtenido de http://www.ugr.es/~rlopezc/sistemas_dialogo.htm
- [5] W3C. (Julio de 2016). *w3c*. Obtenido de
<http://www.w3c.es/Divulgacion/GuiasBreves/Multimodalidad>
- [6] *AndroidZone*. (Junio de 2016). Obtenido de <http://androidzone.org/2013/05/historia-de-android-la-evolucion-a-lo-largo-de-sus-versiones/>
- [7] Xataka. (Septiembre de 2016). *xataka*. Obtenido de xataka:
<http://www.xataka.com/analisis/android-7-0-nougat-analisis-probamos-la-mejor-version-hasta-la-fecha>
- [8] Android Studio FAQs. (Julio de 2016). *Android Studio FAQs*. Obtenido de
<http://androidstudiofaqs.com/conceptos/android-studio-historia>
- [9] Android Studio FAQs. (Julio de 2016). *Android Studio FAQs*. Obtenido de
<http://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos>
- [10] Android Studio FAQs. (Julio de 2016). *Android Studio FAQs*. Obtenido de
<http://androidstudiofaqs.com/conceptos/android-studio-vs-eclipse>
- [11] SGOLIVER. (Agosto de 2016). *SGOLIVER*. Obtenido de
<http://www.sgoliver.net/blog/estructura-de-un-proyecto-android-android-studio/>
- [12] CCM. (Agosto de 2016). *CCM*. Obtenido de <http://es.ccm.net/contents/66-introduccion-bases-de-datos>
- [13] Ub.edu. (Agosto de 2016). *ub.edu*. Obtenido de ub.edu:
<http://www.ub.edu.ar/catedras/ingenieria/Datos/capitulo1/cap14.htm>
- [14] Inesem. (Agosto de 2016). *Inesem*. Obtenido de <http://revistadigital.inesem.es/nuevas-tecnologias/los-gestores-de-bases-de-datos-mas-usados/>
- [15] lab.inf.uc3m. (Septiembre de 2016). *lab.inf.uc3m*. Obtenido de lab.inf.uc3m:
<http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>
- [17] LG. (Septiembre de 2016). *LG*. Obtenido de LG: <http://www.lg.com/es/telefonos-moviles/lg-G3-s-D722>

- [18] microsoftstore. (Septiembre de 2016). *microsoftstore*. Obtenido de microsoftstore:
https://www.microsoftstore.com/store/mseea/es_ES/pdp/productID.324450500?VID=324450600&s_kwcid=AL!4249!3!40280693082!!!g!199197156724!&WT.mc_id=es_datafeed_pla_google&WT.mc_id=pointitsem&WT.mc_id=office&ef_id=VI1@CwAAAR-Mmm33:20160918181014:s
- [19] Aprende a programar. (Julio de 2016). *aprendeaprogramar*. Obtenido de aprendeaprogramar: <http://www.aprendeaprogramarandroid.com/manual-android-studio-principiantes/>
- [#]alt0tip. (Agosto de 2016). *alt0tip*. Obtenido de alt0tip:
<https://www.youtube.com/watch?v=hlQrllf0E6E>
- [#]Android Cero. (Julio de 2016). *Android Cero*. Obtenido de Android Cero:
<http://androidcero.eledevapps.com/2015/01/onclick-en-programacion-android-28.html?showComment=1473786340742#c6032305600900645314>
- [#]Antevenio. (Junio de 2016). Obtenido de <http://www.antevenio.com/blog/2016/01/20-estadisticas-de-marketing-en-2015/>
- [#]Gobierno de España. (4 de Junio de 2016). *exteriores.gob*. Obtenido de exteriores.gob:
http://www.exteriores.gob.es/Portal/es/SalaDePrensa/Multimedia/Publicaciones/Documentos/2014_FOLLETO%20ESPANA%20PAIS%20DE%20TECNOLOGIA%20ESP.pdf
- [#]Pergamino Virtual. (Julio de 2016). *Pergamino Virtual*. Obtenido de
<http://www.pergaminovirtual.com.ar/definicion/InkML.html>
- [20]Fernández, M. D. (2013). *Desarrollo de bases de datos : casos prácticos desde el análisis a la implementación*. Madrid: Ra-Ma.
- [#]Miranda Oliván, A. T. (2004). *Como elaborar un plan de empresa*. Madrid: Thomson Editores Spain.
- [#]Catalán, A. (2011). *Android: Desarrollo de aplicaciones móviles*. Guatemala: Stephanie Falla Aroche.
- [#]Oliver, S. G. (2014). *Curso Programación Android*. Madrid: SGOLIVER.

ANEXO

1. Contenidos en Inglés

1.1 ABSTRACT

The present project has been developed in order to finish the Dual Degree in Computer Engineering and Business Administration. The purpose of this job is relative to the Computer Engineering degree.

This document explains the process to develop a mobile devices app in the education ambit which main aim is to offer the user the functionality to create questions with answers and keywords to finally resolve a test with limited answers or a test with open space to write the answer, when the user completed each test he gets a score.

It has been necessary planning the project with deadlines. The next step was to research the current situation of this topic. In this lines, the investigation was focused in mobile learning platforms, all the apps related with studies and resources to learn are contains in this point. The investigation was centred in similar apps with development in this job with the purpose of value it, their main characteristics what are good for the implementation. Also it was necessary to know more about the available tools in the IT world and which of them are perfect to use, in this case Android Studio to the code and MySQL to create the Data Base were the final election. Everything in this paragraph is the planning part, as the document explains.

When the planning part was finished, it starts the development part. Manuals, tutorials and some videos, what are named in references, were necessary to start the development.

Completed the previous part, the app was tested in 2 platforms: the development tool and a smartphone, with a good assessment.

Before the conclusion and the future investigation will be written, 72 people were interviewed to know if the developed app have the things the public waits and some comments to improve the app. It has tested what the project is well done and the people give some recommendations to improve it.

Finally the conclusion is about the aspects more important of the development and the problems that the developer found. Also, it written a future line of investigation with the comments of the people.

Keywords: mobile learning, Android, operating system, database, test.

1.2 INTRODUCTION

This document exhibit a project from the main idea to it finishes as the purpose of the Final Degree Project. This job is based on the implementation of a mobile app to devices that runs with Android as operating system, and also the user can do the things with 2 interfaces: touch and oral. The current project is divided in 2 parts: first is about the study of similar apps, planning, app design; and the second is about the description of the development, tests, people opinion and general conclusions.

The selected topic for the app is the education and the main goal is help students to test the knowledge that previously they have studied with the tests that they have created with and touch/oral interface.

1.2.1 Motivation

When the project was selected, the motivation to select an app development for a mobile phone was preceded about the technology world. Everything is a click distance and this click is done with the smartphone.

Nowadays, there are 4 big OS: Android, iOS, Windows Phone and BlackBerry. 2 of them have the 98%, they are:

- Android: 75%
- Apple: around 23%

So, 1 each 4 users have an Android mobile. Because of that, it decided Android to offer the app for more people.

After the project was selected, it was the turn to think about what will be the app. While the creation of a game will be hard and there is no a idea to improve the presents games. The election was to implement something to help the users to do traditional actions through their mobile, then appears mobile learning.

1.2.2 Mobile learning

“This educative methodology allows teachers and students be in constant contact every time and everywhere, encouraging with it and individuality education and adapting to the student necessities (everyone are not motivate at the same moment in the day)... and a huge number of advantages.”

Nowadays, technologies have invaded the user's life, from the process more simple as buy food online until the more complex as has a medical met by Skype. Technologies has changed the method to teach and to learn; the resources are online instead of take notes, students doing test online instead of in paper, teachers write mails to help their students, etc.

All of this educative process in which technology are in it can be happen in a space prepared for it, b-learning, or in a space no prepared, e-learning. The student can decide the moment to study and the place to do it. In a lot of times, people have free time and they only have their mobile phone, from a waiting room until a journey in the train. Both are perfect occasions to learn through the mobile.

According to A. Pisanty, L. Enriquez, L. Chaos – Cador, M. García, the mobile learning concept appears in latest 90s in EEUU through PDAs and in Europe appears in 2001 with M learning project which aim was to improve the needed habilities of people without studies, as

mathematics and reading comprehension. Today the project still exists to help people that do not want to receive formal education.

From latest 90s until today, the number of users that have a smartphone have increased exponentially and the first time that people have a mobile phone has reduced a lot. Although a huge number of users only use their mobile to leisure activities, this threat can be converted in a opportunity through the educative tools. The education should adapt to the advances of the society and can convert the mobile problem in the schools in the solution to open resources that not appears in the text book or replace it.

The definition that is more similar to the aim of mobile learning is the e—ISEA offer:

“Mobile learning is a new way of education creating with the combination of e-learning and the use of smartphones and that is based in the possibilities this devices offers to the user. It allow to combine geographic movility with virtual, that allows learn in a context, and in the time is needed, exploring and requesting correct information”

This table is a summary of the main advantages and disadvantages of the use of mobile learning:

Advantages	Disadvantages
Learn everywhere and everytime	Small screens
Easy communication teacher-student	Rejection new technologies
Accesibility to more resources	Distraction of users
Collaborative learning	Store in the device
Motivation of study	High battery consumption
Intregation education – new technologies	Constant updates

The advantages that are possible grateful this technology are infinity, while the disadvantages are about problems with the device that can be solved with external store cards and external batteries.

Mobile learning is the educative process of the future because it offers a huge number of advantages. Nowadays it is impossible to imagine a world without technology and the education should adapt to it. Universities have implanted a system to easy the communication between teacher and students. Schools are implanting it so it is a future project. Students have to use the new system outside the university, for example with the use of mobile apps.

After this research, the app to develop is going to offer to students the possibility of retake their knowledge without carry books and notes everywhere, simply with the app to do some tests with it.

1.3 AIM

This section is divided in 2 parts: project's aims and development app aims.

1.3.1 TFG aim

- Research of similar apps in smartphones: this investigation is done with the goal to find some characteristics of apps with a question-answer structure to know what can offer to the users for they select the development app.
- Learning to use new tools related with the development of mobile apps: with the goal to development the app, learn to use the necessary tools:
 - Android Studio: platform where the code is written.
 - Justinmind: design of interfaces
 - MySQL: creation of a database.
- Use code languages to Android programming: guide the languages studied in the degree to the Android programming: XML, Java and SQL.
- Create an app that can be used in mobile devices: development an app that can be downloaded and uses in smartphones to make test and learn something.

1.3.2 App aim

- Offer to the user a simple and useful option instead of the existing: offer to the user a simple app through an intuitive interface. The app help them to retake their knowledge about topics they want to learn.
- Collaborating in the integration of mobile learning: offer to the user a easy app that help for the integration of the new technologies in the education.
- Interface touch and oral: insert the questions and answers, the reader of them to do the test with an oral and touch interface.

1.4 PLANNING

The production of this project is divided in 3 parts: planning, development and redaction.

1.4.1 Planning phase

Phase of Establishment of aims, project level, develop level and deadlines. Also, it is about the previous study of similar apps and interfaces design. It is explain in the next list:

- Establishment of aim to know the scope of the project and the app.
- Deadlines establishment, it is in a GANTT diagram what is contained in figure 2 of section 1.4 Temporal Planning
- Tool selection to implement the development of the app: Android Studio versus Eclipse with Android plugin. This decision was taken as use tool for the future.
- Research about similar apps: the goal is copy the good functionalities and insert new of them.
- Design of nterfaces: design of the different screens that user can see in the app. Following the aim of use new tools, the app selected was Justinmind.

1.4.2 Development phase

Development of the app that users can be enjoy to do tests of their knowledge. Also of the development, it contains a test phase to probe the app. The development is divided in 4 nuclear tasks:

- Interfaces development in Android Studio with XML tags.
- Business logic development in Android Studio with java knowledge.
- Persistence development in Android Studio and in MySQL with SQL development. Also of implement the different queries in Android Studio, it is necessary to create a Data Base in MySQL, if it is not exists the app cannot save the questions.
- Google dialog system import in the tool Android Studio.

The probes to test the app will be detailed in section Performed probes.

1.4.3 Redaction phase

Phase to write the previous study, app status and problems found by the analyst.

This phase will be do when the development finishes, although it follows the notes which have been taken and the experience in the development.

1.5 NEEDED RESOURCES

In this section it is detailed the software tools and hardware neede to completed the project, the programming languages which have been necessary to the development and manuals/tutorials. In the next lines, this resources are divided in: software, hardware, programming languages and manuals.

1.5.1 Software

- Operative system Windows 10→ platform to install the tools.
- Microsoft office Word 2016→ tool to write the project.
- Justinmind → design of interfaces
- GANTT Project → establishment of deadlines.
- Android Studio→ development of interfaces, business logic and persistence.
- MySQL→ creation of a Data Base
- Notepad ++ → visualizer of code.

1.5.2 Hardware

- Laptop Lenovo ideapad y410p→ device to prepare the workspace
- Smartphone LG G3S→ app test.

1.5.3 Programming languages

- XML→ development of interfaces
- Java →development of business logic
- SQL → development of persistence and database

1.5.4 Manuals

- Project Adrián Muñoz Moreno ADE → project structure and how to do a budget
- Manual “Curso Android Studio”→how to programme in Android
- Different online websites to know how to do something in the develop

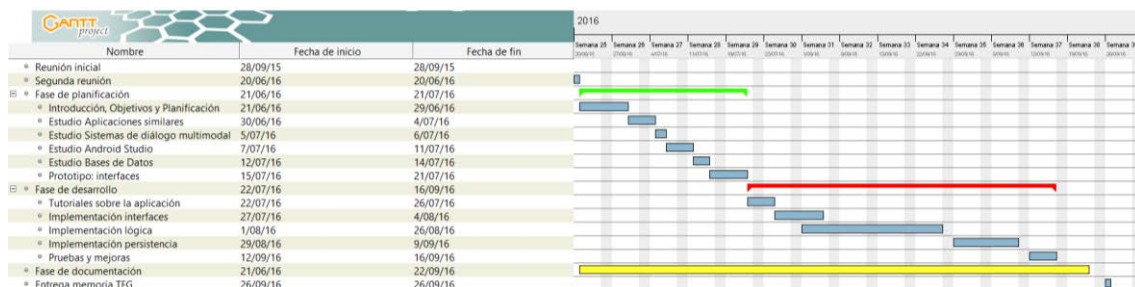
1.6 Temporal planning

The main idea of the project was born in September of 2015 when the student selected the project “DESARROLLO DE UN ASISTENTE MULTIMODAL EDUCATIVO PARA DISPOSITIVOS MÓVILES ANDROID”. After the first meeting, it was on 29/09/2015 it specify that the app will be a question test app in which the user can include her own questions and resolve some tests.

The first meeting was in September because the project should be selected and registered in the first assessment, although of that and because the student have subjects in the 2 assessment and the business project, what is was presented in July of 2016, the computer engineering project was postpone until summer. Because of that the GANNTT diagram goes from 1/06 to 26/09 of 2016.

The first step was a previous study about the ambit of the app. The next step was the design of interfaces and the structure of the app. Then, the development starts and when it finishes, the project finish with this document.

The GANTT diagram is here:



1.7 Budget

The next table contains the hour cost of each worker that is implicated in the project and the number of hours they are going to work in it. In the last column, it is the final cost of each worker. To know the hour cost, it have done a research on the internet, and to estimate the number of hours of each worker it has revised the GANTT diagram and the hours invested in the development.

Category	Cost per hour(€)	Hours	Cost
Project Manager	40,00 €	40	1.600,00€
Analyst	25,00 €	150	3.750,00€
Programmer	20,00 €	250	5.000,00€
Quality Manager and testing	15,00 €	40	600,00€
Total		480	10.950,00€

In the next table it is the budget of the devices and tools necessary to the development:

Device/Tool	Cost	Useful life (months)	Use time (months)	Chargeable cost
Laptop lenovo ideapad y410p	908,99€	36	3	75,749€
Smartphone LG G3S	121,18€	24	3	15,148€
Windows Office package	149,00 €	12	3	37,25€
Android Studio	0,00€	--	3	0,00€
MySQL Workbench	0,00€	--	3	0,00€
JustInMind (trial version)	0,00€	--	3	0,00€
Register Play Store	22,40€	--	--	22,40
			Total	150,547€

For the devices and tools it necessary to charge to the project the part of the amortized costs in its use because it is not exclusive for the project. It is selected a lineal amortization for its chargeable, the devices and tools will be amortized in the same quantity every year of its useful life:

$$\text{Chargeable cost} = \frac{\text{Cost}}{\text{Useful life}} \times \text{use time}$$

With personal costs and device/tool cost, it obtained the total costs table:

Origin of the cost	Coste
Personal	10.950,00€
Device/tools	150,547€
Total	11.078,147€

The way to get profits with this project will be through the sales of the app in Play Store. The app will cost 1, 00€, 30% of it will be to Google, so the development team will earns 0, 70 with each sale. The dead point in which the initial cost is covered is about 11.078, 147€ or 15.825 downloads.

1.8 DOCUMENT STRUCTURE

In the next lines, it is include a summary of each chapter in this document to the reader know the structure of it:

1. Introduction

This point specify how is the project, the motivation of the student, aim, planning establishment, resources needed and the budget necessary.

2. State of art

It explain the similar apps, advantages and disadvantages of them, a research about multimodal systems. Also a investigation of databases and it is selected one of them.

3. General description about the development app

The screens of the app and their functionalities are showed. The architecture in the development and the relational schema for the database.

4. App assessment

It specify the 11 tests that the app have to pass. Also, it is include a interview to people for know their opinion about the app.

5. **Conclusion and future lines of investigation**

An analysis of the obtained results and it is exposed a conclusion as a summary. It is announced a future line of investigation to improve the project.

6. **Glossary**

Summary explain about the non-natural words.

7. **References**

It contains the multiples websites and books consulted in the development of the project.

8. **Annexed**

It contains all the important information of the project that is not in the previous chapter as the designed prototype.

1.9 GENERAL CONCLUSIONS

This project is about the investigation of an app development to mobile devices which OS is Android. The user can use a touch or oral interface. The app allows the users to test their knowledge through test questions or development questions which answers the user previously created in the app. The user can navigate in the screens through the touch interface and he can resolve the questions with the touch or oral interface.

The main task consisted in define the scope of the app and of the document. The beginning was started after the date that the planning established so the rest of the dates were delayed from the initial planning. Also the development phase needed more time than the established so the analyst have learned to organized better the deadlines and in a way with more order, establishing date with a margin for incidentals.

Initially, it was studied similar apps and the summary is that users want a simple and intuitive interface that allows them to test their knowledge. Without strong colours and with basics functionalities.

The development phase was the most complicated because there were a lot of problems with the download of the main tool for the implementation. The main problems are that the necessary SDK that allows to the tool starts was wrong and some paths with an accent mark don't allow save the software. Although, grateful is a very used tool, the level of information online is high in forums so the problems can be solved.

With the workspace prepared, the development starts with the design of the interfaces follow the prototype and with video tutorials online. This sub phase don't needed much time. After of that, it started the business logic development, first of all the interface controls of the presentation layer, were linked with the java code of the business logic layer. To complete this task was necessary consult forums, although was a tedious task because of the high number of controls for each screen, the task was converted in an iterative process.

The next step was the most complicated task of the project: implement the business logic layer. It should programming the nuclear layer what communicates the information contained in the database with the app screens. First of it was established a connector for send orders from this layer to persistence layer and there were a lot of problems to get it. It filled more time than the estimated for it.

At the same time, it was create a database in MySQL tool where the information will be saved, this task was simple because of through the courses in the university it was something iterative. Once the DB was running, the queries were created with variable fields what will be filled with the user information.

Finally, the redaction phase do not finish until the last minute to write everything about the app development in a clear and formal format. In this phase, the analyst tested the app with good results and some users were interviewed what reveals that the development functionalities are useful for the potential clients, also they write some reviews to improve the app.

1.9.1 FUTURE LINES OF INVESTIGATION

In the next lines there are exposed the future lines of investigation of this project:

- The main necessity is programming this app in other OS such as iOS. First of all, it is going to be research information about apple.
- The best opinions of the interviewed user will be implemented to complete the app:
 - Option to save notes.
 - Links to the questions.
 - Historic failed answers.
 - Option to take notes and make schemas.
 - Change strong colours.
 - Field to include the difficult of the question.
 - Option to upload images.
 - Social integration: chat with other users, reviews, help section, results sharing, and users rank.
 - Roles to difference students and teachers.
- Option to change the language to offer the app in other countries.
- Option to change the colour of the app.
- Improve development question because it only contains 6 keywords and it runs if the word is in it without know the context.

2. Prototipo JustinMind

Justinmind es una herramienta que permite el modelado de interfaces. Para ello, se selecciona un dispositivo entre los disponibles y se van moviendo contenedores hasta él, pudiendo cambiar el color de dichos contenedores y las etiquetas que se muestran. Para el desarrollo de las interfaces de la aplicación móvil desarrollada, se escogió esta herramienta por su sencillez y la gama de funcionalidades que ofrecía. En el prototipo diseñado, se pueden observar las distintas pantallas y cómo funcionaría la navegación entre ellas.

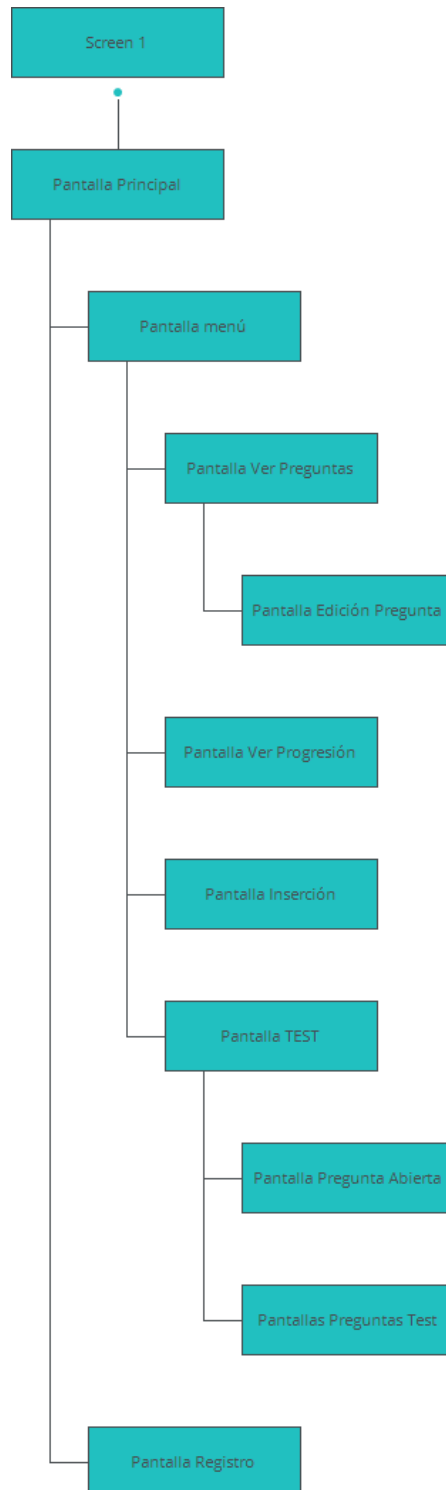


Figura 34. Esquema pantallas del prototipo

01 - Screen 1

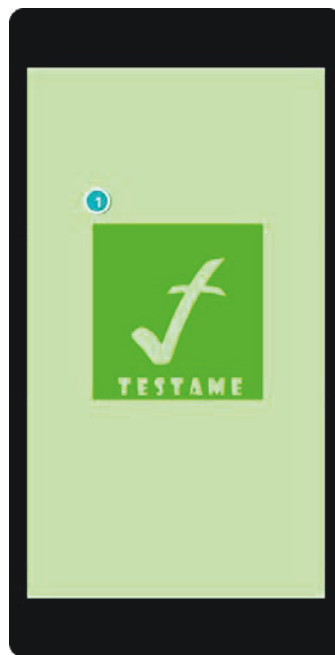


Figura 35. Pantalla I Prototipo

1 on Click: goes to 'Pantalla Principal' →

02 - Pantalla Principal

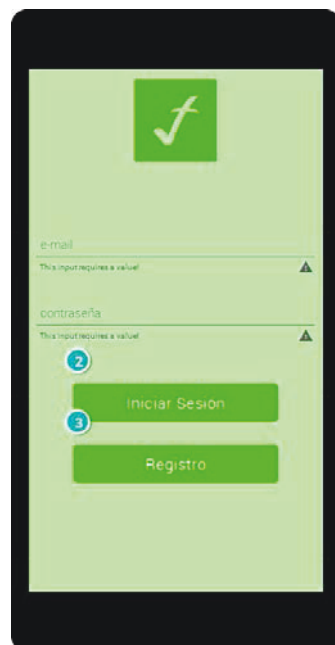


Figura 36. Pantalla II Prototipo

2 on Click: goes to 'Pantalla menú' →

3 on Click: goes to 'Pantalla Registro' →

03 - Pantalla Registro

A mobile app registration screen with a light green background. At the top is a green square icon with a white checkmark. Below it are four input fields: 'Nombre de usuario', 'Dirección de e-mail', 'Confirme su dirección de e-mail', and 'Contraseña'. The email fields contain the placeholder 'ejemplo@dominio.com'. The password field has a masked input with dots. A blue circle with the number '9' is positioned to the left of a green 'Registrarme' button at the bottom.

Figura 37. Pantalla III Prototipo

9 on Click: goes to 'Pantalla menú' →

04 - Pantalla menú

A mobile app menu screen with a light green background. At the top is a green square icon with a white checkmark and a blue circle with the number '8' to its left. Below the icon are two dropdown menus: 'Seleccione una materia' and 'Añade Tema', with a green '+' button to the right of the second dropdown. Below these are five green buttons stacked vertically: 'TEST', 'Insertar Pregunta', 'Ver Preguntas', 'Ver Progresión', and 'Ver Preguntas'. Each button has a blue circle with a number to its left: '7' for 'TEST', '6' for 'Insertar Pregunta', '4' for 'Ver Preguntas', '5' for 'Ver Progresión', and '4' for 'Ver Preguntas'.

Figura 38. Pantalla IV Prototipo

4 on Click: goes to 'Pantalla Ver Preguntas' →

5 on Click: goes to 'Pantalla Ver Progresión' →

6 on Click: goes to 'Pantalla Inserción' →

7 on Click: goes to 'Pantalla TEST' →

8 on Click: goes to 'Pantalla Principal' →

05 - Pantalla TEST

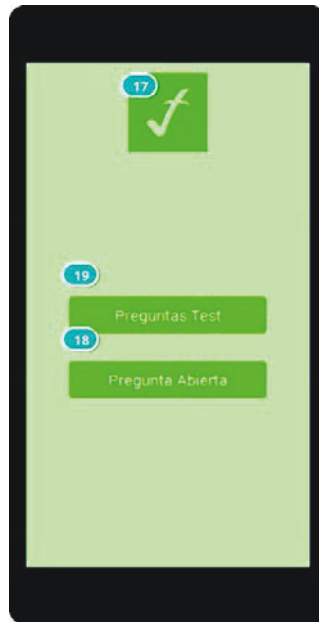


Figura 39. Pantalla V Prototipo

- 17 on Click: goes to 'Pantalla Principal' →
- 18 on Click: goes to 'Pantalla Pregunta Abierta' →
- 19 on Click: goes to 'Pantallas Preguntas Test' →

06 - Pantallas Preguntas Test

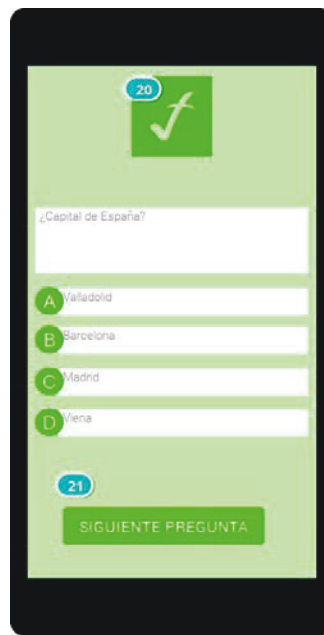


Figura 40. Pantalla VI Prototipo

- 20 on Click: goes to 'Pantalla Principal' →
- 21 on Click: goes to 'Pantalla menú' → goes to 'Pantalla Inserción' →

07 - Pantalla Pregunta Abierta

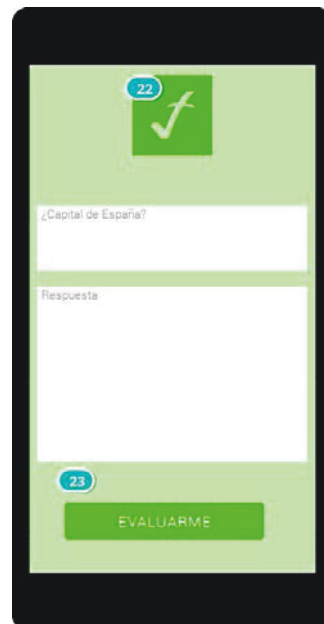


Figura 41. Pantalla VII Prototipo

- 22 on Click: goes to 'Pantalla Principal' →
- 23 on Click: goes to 'Pantalla menú' → goes to 'Pantalla Inserción' →

08 - Pantalla Inserción

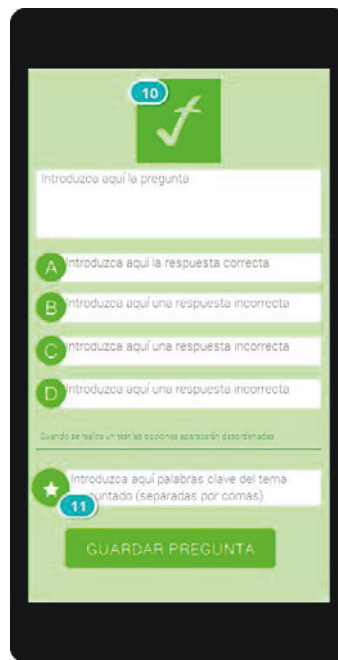


Figura 42. Pantalla VIII Prototipo

10 on Click: goes to 'Pantalla menú' →

11 on Click: goes to 'Pantalla menú' → goes to 'Pantalla Inserción' →

09 - Pantalla Ver Preguntas

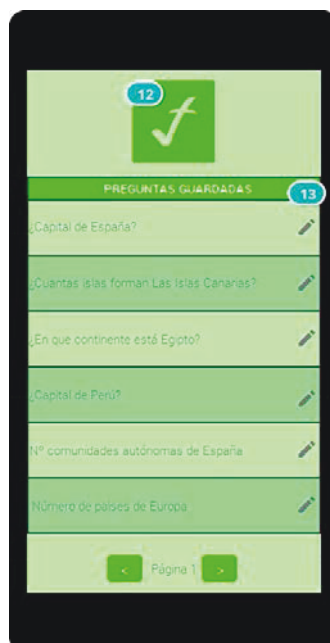


Figura 43. Pantalla IX Prototipo

12 on Click: goes to 'Pantalla Principal' →

13 on Click: goes to 'Pantalla Edición Pregunta' →

10 - Pantalla Edición Pregunta



Figura 44. Pantalla X Prototipo

- 14 on Click: goes to 'Pantalla menú' →
- 15 on Click: goes to 'Pantalla menú' → goes to 'Pantalla Inserción' →

11 - Pantalla Ver Progresión

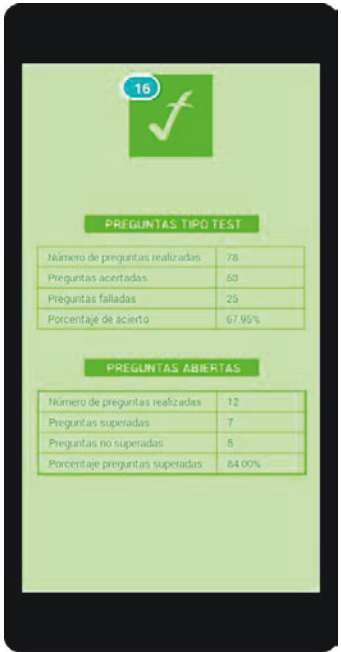


Figura 45. Pantalla XI Prototipo

- 16 on Click: goes to 'Pantalla Principal' →